
Preprint No. M 02/08

**Bandbreitenreduktion -Teil 3 -
Algorithmus von Gibbs-Poole-
Stockmeyer- Testbeispiele mit CM
und GPS**

Neundorf, Werner

September 2002

Impressum:

Hrsg.: Leiter des Instituts für Mathematik
Weimarer Straße 25
98693 Ilmenau
Tel.: +49 3677 69 3621
Fax: +49 3677 69 3270
<http://www.tu-ilmenau.de/ifm/>

ISSN xxxx-xxxx

ilmedia

Technische Universität Ilmenau
Fakultät für Mathematik
und Naturwissenschaften
Institut für Mathematik

http://www.mathematik.tu-ilmenau.de/Math-Net/js/home_de.html

Postfach 10 05 65
D - 98684 Ilmenau
Germany
Tel.: 03677/69 3267
Fax: 03677/69 3272
Telex: 33 84 23 tuil d.
email: werner.neundorf@tu-ilmenau.de

Preprint No. M 08/02

Bandbreitenreduktion - Teil 3

Algorithmus von Gibbs-Poole-Stockmeyer
Testbeispiele mit CM und GPS

Werner Neundorf

September 2002

[‡]MSC (2000): 65F50, 65F05, 65-01, 65-04, 65-05, 68Q25

Zusammenfassung

Gegenstand des dreiteiligen Preprints ist die Bandbreitenreduktion von Matrizen. Seine einzelnen Ausgaben basieren auf dem Vorlesungsskript "Wissenschaftliches Rechnen - Matrizen und LGS", gehalten als fakultative Veranstaltung am Institut für Mathematik der TU Ilmenau.

Teil 1 enthält die Grundlagen dazu, die insbesondere auf sparse und Bandmatrizen und deren Verarbeitung eingehen. In den Teilen 2 und 3 werden die Bandbreitenreduktion mit dem Algorithmus von Cuthill-McKee bzw. Gibbs-Poole-Stockmeyer ausführlich erläutert, verglichen und an Beispielen illustriert.

Vorwort

Viele Probleme benötigen die Handhabung von Matrizen bzw. die Lösung von linearen Gleichungssystemen. Ausgangspunkt dabei ist, dass man alle bzw. wichtige Informationen über die Matrix nutzt und diese auf ihre weitere Verarbeitung "vorbereitet".

Zu solchen Maßnahmen gehören:

- Feststellung von Eigenschaften der Matrix in Bezug auf Symmetrie, (strenge) Regularität, Definitheit, Diagonaldominanz, Orthogonalität u. a.,
- Erkennen und Anwendung der Besetztheitsstruktur,
- Bandbreiten- und Profilverzögerung,
- (symmetrische) Zeilen/Spaltenpermutation,
- Elementeabgleich,
- Zerlegungs- und Transformationstechniken.

Dabei liegen die Untersuchungen in folgenden Problemklassen.

- (1) Skalierung als eine Form der Verbesserung der Kondition der Matrix.
- (2) Faktorisierungsmethoden der Form $A = BC$, $A = BCD$ oder ähnlich unter Einbeziehung von Aspekten, die sie numerisch gutartig machen.
Damit ist natürlich formal unter zusätzlichen Bedingungen eine Transformation $C = B^{-1}A$ beschrieben.
- (3) Transformationsmethoden der Form $A' = BAC$ möglichst mit Angabe der Transformationsmatrizen B und C , einschließlich der Betrachtung von Sonderfällen.
Ziel dabei ist es, dass die transformierte Matrix A' Eigenschaften besitzt, die ihre weitere Nutzung effizienter machen.

Hier soll der Schwerpunkt auf den Anstrich (3) gelegt werden.

Dazu werden Lösungsalgorithmen bzw. implementierte Routinen in der verschiedenen Programmiersprachen oder Computeralgebrasystemen angegeben.

Einige ergänzende grundlegende Abschnitte sowie zahlreiche Beispiele sollen insgesamt das Verständnis für die Problematik unterstützen.

Inhaltsverzeichnis

1	Bandbreitenreduktion	1
1.1	Algorithmus von Gibbs-Poole-Stockmeyer	6
1.1.1	Grundlagen	7
1.1.2	Programmerläuterungen zu GPS	29
1.2	Testbeispiele zur Bandbreitenreduktion	50
	Literaturverzeichnis	75

Kapitel 1

Bandbreitenreduktion

Für LGS mit Bandmatrizen gibt es spezielle, meist effizientere Lösungsverfahren. In vielen Fällen lässt sich die Bandbreite einer Matrix $A = (a_{ij})$ durch simultane Zeilen- und Spaltenvertauschungen erheblich reduzieren. Die Transformation ist

$$A' = (a'_{ij}) = P^T A P, \quad P P^T = P^T P = I, \quad P^T = P^{-1}, \quad (1.1)$$

wobei P eine Permutationsmatrix ist, die durch die Permutation $p = (p_1, p_2, \dots, p_n)$ (Permutationsvektor) erzeugt wird. Die Matrix P hat in ihrer i -ten Zeile an der j -ten Spalte mit $j = p_i$ eine Eins stehen, sonst Nullen.

Eine solche simultane Vertauschung von Zeilen und Spalten ist also eine Ähnlichkeitstransformation.

Einige Beispielmatrizen sollen auf diese Weise transformiert werden. Dazu sind Bandbreite bw und Anzahl nne der Nichtnullelemente (NNE) angegeben ([64], Kap. 1.3).

Beispiele

(a)

$$A = (a_{ij}) = \begin{pmatrix} 2 & & 1 & & \\ & 3 & & & 1 \\ & 2 & & 1 & \\ & & 1 & 2 & \\ & & 1 & 2 & 1 \end{pmatrix}, \quad bw(A) = 6, \quad nne = 11.$$

$$p = (1, 5, 2, 3, 4), \quad A' = P^T A P = \begin{pmatrix} 2 & 1 & & & \\ 2 & & 1 & & \\ & 1 & 2 & & \\ & 1 & 2 & 1 & \\ & & & 1 & 3 \end{pmatrix}, \quad bw(A') = 4.$$

(c) Wir verringern die Bandbreite der symmetrischen (60×60) -Matrix aus dem m-File `bucky` der Demo-Toolbox von MATLAB.

```
>>B = bucky;
```

```
>>spy(B)
```

```
>>p = symrcm(B)
```

```
p =
```

1	6	2	5	10	11	12	7	26	30	3	4
9	15	13	8	27	29	16	17	21	25	38	14
37	42	28	43	20	18	22	24	39	33	36	41
47	44	19	32	23	48	40	34	45	46	53	31
52	49	57	35	58	50	54	51	56	59	55	60

```
>>spy(B(p,p))
```

```
>>print bild08.ps -dps
```

```
>>[i,j] = find(B)
```

```
>>bw = max(i-j)-min(i-j)+1
```

```
bw =
```

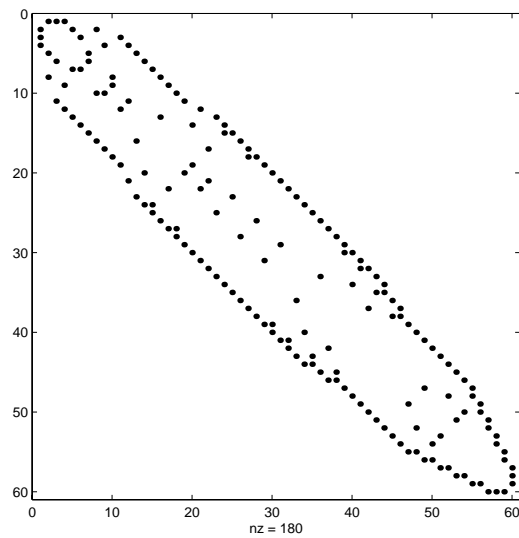
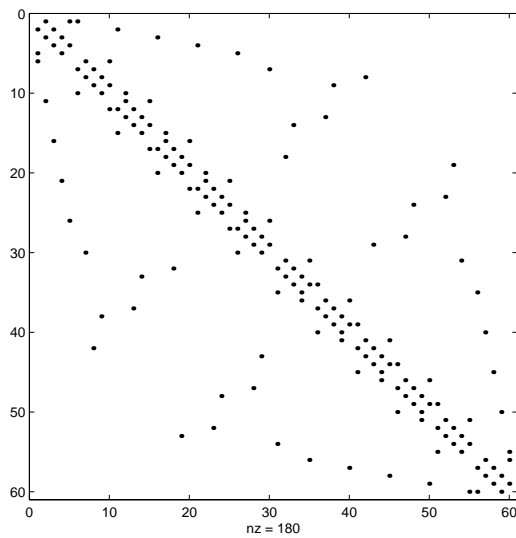
```
69
```

```
>>[i,j] = find(B(p,p))
```

```
>>bw = max(i-j)-min(i-j)+1
```

```
bw =
```

```
21
```



(d)

$$A = \begin{pmatrix} * & & * \\ & * & \\ & & * \\ & & & * \\ & & & & * \end{pmatrix}, \quad bw(A) = 5, \quad b1(A) = 9, \quad nne = 6.$$

$$p = (1, 5, 3, 4, 2), \quad A' = P^T A P = \begin{pmatrix} * & * & & & \\ & * & & & \\ & & * & & \\ & & & * & \\ & & & & * \end{pmatrix}, \quad bw(A') = 2, \quad b1(A') = 3.$$

(e)

$$A = \begin{pmatrix} * & & * \\ & * & \\ & & * \\ & & & * \\ * & & & & * \end{pmatrix}, \quad bw(A) = b1(A) = 9, \quad nne = 7.$$

$$p = (1, 5, 3, 4, 2), \quad A' = P^T A P = \begin{pmatrix} * & * & & & \\ * & * & & & \\ & & * & & \\ & & & * & \\ & & & & * \end{pmatrix}, \quad bw(A') = b1(A') = 3.$$

(f)

$$A = \begin{pmatrix} * & * & & & * \\ * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * & * \\ * & & & & * & * \end{pmatrix}, \quad bw(A) = b1(A) = 11, \quad nne = 18.$$

$$p = (1, 3, 5, 6, 4, 2), \quad A' = P^T A P = \begin{pmatrix} * & * & * & & & \\ * & * & & * & & \\ * & & * & & * & \\ & * & & * & & * \\ & & * & & * & * \\ & & & * & * & * \end{pmatrix}, \quad bw(A') = b1(A') = 5.$$

Eine weitere Reduktion der Bandbreite ist nicht möglich.

Zu jeder Matrix A gibt es eine Permutation, die A in eine Matrix mit minimaler Bandbreite überführt. Diese Permutation ist in der Regel nicht eindeutig bestimmt. Leider gibt es für große n keinen Algorithmus, der eine Permutation dieser Art mit vertretbarem Rechenaufwand (polynomialer Aufwand) bestimmt. Die praktisch benutzten Algorithmen liefern Permutationen, die im Regelfall zu einer Bandbreite in der Nähe der minimalen Bandbreite führen. Diese Feststellung beruht auf praktischen Erfahrungen. Es lässt sich nur schwer theoretisch voraussagen, wie weit die erreichte Bandbreite von der minimalen abweicht.

Von zahlreichen approximativen Algorithmen dieser Art werden der von Cuthill-McKee (CM) 1969 [4], [56], [57], der umgekehrte Cuthill-McKee (reverse Cuthill-McKee, RCM) sowie eine Modifikation von Gibbs-Poole-Stockmeyer (GPS) 1976 [29] am meisten benutzt. Beide Algorithmen beruhen auf graphentheoretischen Überlegungen. Manchmal liefert der erste die besseren Ergebnisse, manchmal der letzte. Die Unterschiede sind meistens gering. Der Algorithmus von Gibbs-Poole-Stockmeyer erfordert fast immer weniger Rechenzeit.

1.1 Algorithmus von Gibbs-Poole-Stockmeyer

Im GPS wird eine weitere Verbesserung von CM [65] bez. Bandbreite und Profil durch die Maximierung der Anzahl der Stufen sowie damit durch die Minimierung der mittleren Anzahl von Knoten pro Stufe erreicht.

Gibbs, Poole und Stockmeyer haben vorgeschlagen, zuerst einen sogenannten Durchmesser des Graphen zu bestimmen, der durch zwei Knoten festgelegt wird, deren kürzester Verbindungsweg im Graphen am längsten ist. Seine Bestimmung erfolgt mit Hilfe des CM-Algorithmus.

Für einen Startknoten v , etwa mit minimalen Grad, wird die zugehörige Stufenstruktur ermittelt. Mit allen Knoten der letzten Stufe wiederhole man den Prozess. Falls dabei eine tiefere Stufenstruktur gefunden wird, übernimmt der betreffende Knoten die Rolle von v , für den die Stufenstruktur bereits bekannt ist. Andernfalls wählt man unter Knoten der letzten Stufe denjenigen u aus, für welchen die maximale Anzahl von Knoten pro Stufe, also Stufenbreite, am kleinsten ist. Die so ermittelten Knoten u und v bilden nicht mit Sicherheit die Endpunkte eines Durchmessers, doch wird ihre Distanz annähernd maximal sein, und man begnügt sich mit einem Pseudodurchmesser.

Dieser erste Schritt des GPS ist sehr geeignet, automatisch günstige Startknoten für den CM oder RCM zu bestimmen, da der Aufwand relativ gering ist und in der Regel nur wenige Startpunkte getestet werden müssen, um einen Pseudodurchmesser zu finden.

In vielen Anwendungsbeispielen können aber die erfolgversprechenden Startknoten mit einiger Überlegung oder auf Grund von Erfahrungen leicht angegeben werden, für welche die Nummerierungen zu ermitteln sind.

In einem zweiten Schritt sollen die beiden Stufenstrukturen, welche zu den beiden Startpunkten u und v auf einem (Pseudo-) Durchmesser gehören, zu neuen Stufenstruktur kombiniert werden. Dazu bilden die Durchschnitte entsprechender Stufen einen ersten Kern der neuen Stufen, die anschließend nach bestimmten Auswahlkriterien mit den noch nicht einbezogenen Knoten ergänzt werden, dass jede Stufe schließlich eine möglichst kleine Zahl von Knoten enthält.

Zum Schluss werden die Knoten innerhalb der neuen Stufenstruktur in Anlehnung an den CM mit gewissen Modifikationen durchnummeriert, da jetzt eine Stufe Knoten enthalten kann, die nicht mehr zu einem Knoten der vorangehenden Stufe benachbart sind. Die oben erwähnte, aus der CM- oder RCM-Nummerierung resultierende tridiagonale Blockstruktur der zugehörigen Matrix B geht natürlich verloren.

Der GPS soll nun genauer erläutert werden.

1.1.1 Grundlagen

Einige graphentheoretische Grundlagen wurden bereits in [65], Kap. 1.1, vermittelt. Wir nehmen an, dass die Matrix A symmetrisch ist. Gemäß [64], Abschnitt 1.3, geben wir ihre Bandbreiten an.

Ihre gewöhnliche Bandbreite bw ist

$$bw(A) = 2\alpha + 1, \quad (1.2)$$

wobei

$$\alpha = \begin{cases} 0, & \text{falls } a_{ij} = 0 \text{ für alle } i > j \\ \max_{\substack{i > j \\ a_{ij} \neq 0}} (i - j), & \text{sonst,} \end{cases}$$

Für die zweite modifizierte Bandbreite $b2$ gilt damit

$$b2(A) = \min(m, \text{ wobei } a_{ij} = 0 \forall i, j \text{ mit } |i - j| > m) = \alpha. \quad (1.3)$$

Die Diagonalelemente von A haben keine Bedeutung für die Bandbreite. Sie seien deshalb wie auch die sonstigen NNE der Matrix o.B.d.A. alle 1.

Die Bandbreiten von

$$A = (a_{ij}) \quad \text{und} \quad B = (b_{ij}) = (|a_{ij}| + |a_{ji}|)$$

stimmen damit überein.

Wenn P eine beliebige Permutationsmatrix ist, gilt das gleiche für die Bandbreiten von

$$P^T A P \quad \text{und} \quad P^T B P$$

Die i -te und j -te Zeile von A heißen verbunden (direkt verbunden, benachbart), wenn $a_{ij} \neq 0$ ist.

Wir schreiben

$$z_i \sim z_j \quad \text{oder} \quad z_j \sim z_i. \quad (1.4)$$

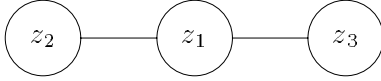
Die Verbindungen der Diagonalelemente $z_i \sim z_i$ zieht man nicht in Betracht. Diese würden bei den daraus sich ableitenden Graphen zu Schleifen führen.

Diese Verbindungen gestatten die Konstruktion eines ungerichteten Graphen mit nummerierten Knoten z_i oder einfach i . Die Zeilen (Zeilennummern) der Matrix sind also die Knoten des Graphen. Die Anzahl seiner Kanten ist gleich der Anzahl n_{neo} der NNE oberhalb der Diagonalen.

Beispiel 1.1

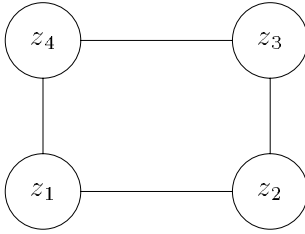
(1)

$$A = \begin{pmatrix} * & * & * \\ * & * & \\ * & & * \end{pmatrix}, \quad nne = 7, \quad nneo = 2$$

Verbindungen: $z_1 \sim z_2, z_1 \sim z_3$ Ungerichteter Graph, $V(G) = \{z_1, z_2, z_3\}$ 

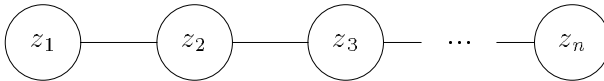
(2)

$$A = \begin{pmatrix} * & * & & * \\ * & * & * & \\ & * & * & * \\ * & & * & * \end{pmatrix}, \quad nne = 12, \quad nneo = 4$$

Verbindungen: $z_1 \sim z_2, z_1 \sim z_4, z_2 \sim z_3, z_3 \sim z_4$ Ungerichteter Graph, $V(G) = \{z_1, z_2, z_3, z_4\}$ 

(3)

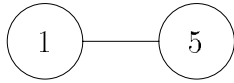
$$A(n, n) = \begin{pmatrix} * & * & & & \\ * & * & * & & \\ & \ddots & \ddots & \ddots & \\ & & * & * & * \\ & & & * & * \end{pmatrix}, \quad nne = 3n - 2, \quad nneo = n - 1$$

Verbindungen: $z_1 \sim z_2, z_2 \sim z_3, \dots, z_{n-1} \sim z_n$ Ungerichteter Graph, $V(G) = \{z_1, z_2, \dots, z_n\}$ 

Umgekehrt lässt sich aus dem Graphen das Muster der Matrix sofort rekonstruieren.

Beispiel 1.2

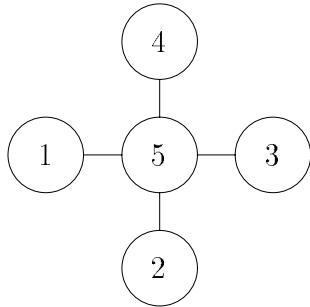
- (1) Der ungerichtete Graph G , $V(G) = \{z_1, z_2, z_3, z_4, z_5\}$



führt mit $n = 5$ auf die Matrix

$$A = \begin{pmatrix} * & & & & * \\ & * & & & \\ & & * & & \\ & & & * & \\ * & & & & * \end{pmatrix}.$$

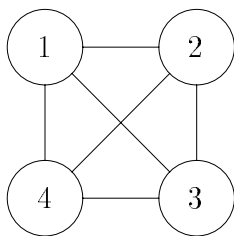
- (2) Der ungerichtete Graph G , $V(G) = \{z_1, z_2, z_3, z_4, z_5\}$



führt mit $n = 5$ auf die Matrix

$$A = \begin{pmatrix} * & & & & * \\ & * & & & * \\ & & * & & * \\ & & & * & * \\ * & * & * & * & * \end{pmatrix}.$$

- (3) Der ungerichtete Graph G , $V(G) = \{z_1, z_2, z_3, z_4\}$



führt mit $n = 4$ auf eine vollbesetzte Matrix.

(G, \sim) , $G = (V, E)$, $V = V(G)$, $E = E(G)$, heißt **ungerichteter Graph**, wenn Folgendes gilt:

- (1) $V(G)$ ist eine nichtleere Menge. Ihre Elemente nennt man Knoten.
- (2) $E(G)$ ist eine nichtleere Menge von Kanten. Sie ist eine Teilmenge aller möglichen Kanten.
- (3) \sim ist eine zweistellige Relation, die zwischen gewissen Knoten g und h besteht. Wir schreiben: $g \sim h$ oder g und h sind verbunden (direkt verbunden, adjazent).
- (4) Für alle $g \in V(G)$ gilt $g \sim g$. $g \sim h$ impliziert stets $h \sim g$.

Ein Knoten g hat den Grad (Valenz) l , wenn g mit genau l Knoten in G verbunden ist, also von diesem Knoten l Kanten ausgehen. Manchmal wird dafür auch der Grad $l + 1$ genommen, weil ja insbesondere $g \sim g$ gilt. Aber diese Größe ist insofern problematisch, weil mit der Zulassung der Schleife, die vom Knoten weg- und wieder hinführt, eigentlich der Grad um 2 zu erhöhen ist.

Ein Graph heißt zusammenhängend, wenn es zu beliebigen Knoten g und h aus $V(G)$ stets Knoten $k_i \in V(G)$, $i = 0(1)r$, $r \in \mathbb{N}^+$, mit folgenden Eigenschaften gibt:

- (1) $k_0 = g$, $k_r = h$,
- (2) $k_{i-1} \sim k_i$ für $i = 1(1)r$.

Damit ist ein Weg von g nach h beschrieben. Die Länge dieses Weges ist die Anzahl der dabei durchlaufenen Kanten und beträgt somit r .

Der Abstand zweier Knoten g und h ist ein Weg mit kürzester Länge und wird mit $d(g, h)$ bezeichnet. Aus allen möglichen Abständen von Knoten eines Graphen ergibt sich sein Durchmesser

$$\text{diam}(G) = \max_{g, h \in V(G)} d(g, h). \quad (1.5)$$

Es sei \tilde{G} eine beliebige nichtleere Teilmenge von G .

Dann ist (\tilde{G}, \sim) ebenfalls ein Graph und Untergraph von (G, \sim) . Wenn (\tilde{G}, \sim) zusammenhängend ist, nennen wir kurz \tilde{G} zusammenhängend.

Die Knoten eines endlichen Graphen (G, \sim) seien irgendwie nummeriert, d. h. es sei $V(G) = \{g_1, g_2, \dots, g_n\}$. Zuweilen notiert man einfach $G = \{g_1, g_2, \dots, g_n\}$.

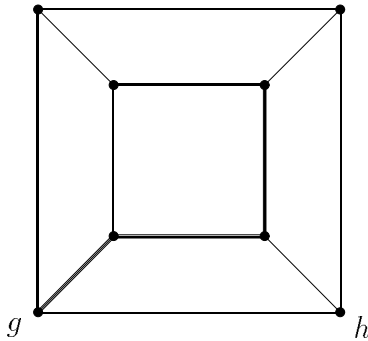
Analog zur Bandbreite einer Matrix heißt

$$w(G) = 1 + 2 \max(d \mid d = |i - j| \text{ mit } g_i \sim g_j) \quad (1.6)$$

dann Bandbreite des Graphen (G, \sim) bezüglich der gegebenen Nummerierung.

Beispiel 1.3 Betrachten wir den Würfelgraphen.

Er hat 8 Knoten und 12 Kanten. Seine Projektion in die Ebene ist

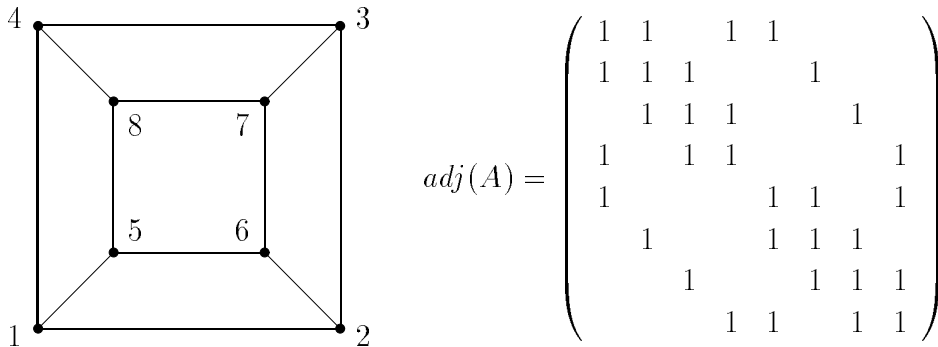


Er hat folgende Merkmale.

- Der Graph ist zusammenhängend.
- Jeder Knoten hat den Grad (Valenz) 3.
- Der maximale Grad ist $D = \max_{g \in V(G)} d(g) = 3$.
- Die Knoten g und h sind verbunden.
- Es gibt mehrere Wege von g nach h , solche mit der Länge 1, 3, 5 usw.
- Der Abstand $d(g, h)$ ist die Länge des kürzesten Weges und beträgt 1.
- Der Durchmesser $\text{diam}(G) = 3$ und ist fett hervorgehoben.

Weitere Eigenschaften ergeben sich bei einer Nummerierung der 8 Knoten.

Die erste Knotenbezeichnung und die zugehörige Adjazenzmatrix seien



Damit sind u. a. $1 \sim 5$, $2 \sim 6$, $3 \sim 7$ und $4 \sim 8$. Die größte Differenz benachbarter nummerierter Knoten ist also 4, so dass in der Matrix 4 obere resp. untere Nebendiagonalen mit NNE entstehen.

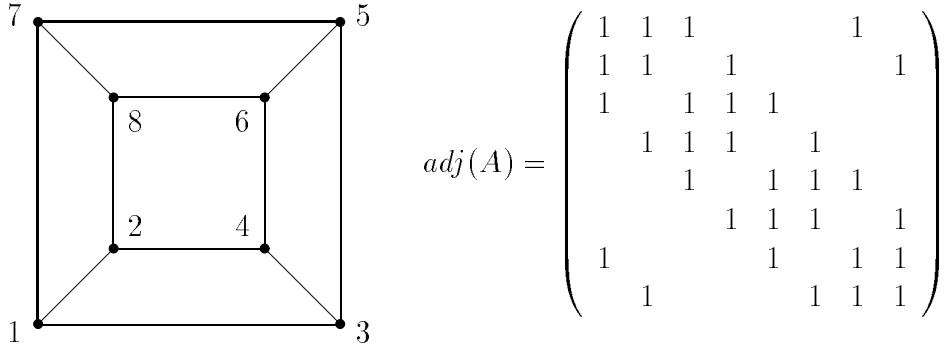
Man erhält für die verschiedenen Bandbreiten

$$b2(\text{adj}(A)) = \alpha = 4 \geq (D + 1)/2 = 2,$$

$$bw(\text{adj}(A)) = 1 + 2 \cdot b2 = 9 \geq D + 1 = 4,$$

$$w(G) = 9.$$

Eine andere paarweise umlaufende Knotenbezeichnung und die zugehörige Adjazenzmatrix sind



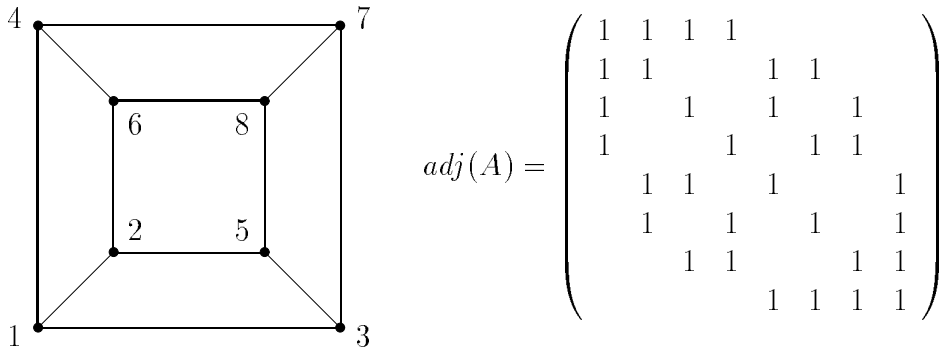
Damit sind u. a. $1 \sim 7$ und $2 \sim 8$. Die größte Differenz benachbarter nummerierter Knoten ist also 6, so dass in der Matrix 6 obere resp. untere Nebendiagonalen mit NNE entstehen.

Man erhält hier für die verschiedenen Bandbreiten

$$b2(adj(A)) = \alpha = 6 \geq (D+1)/2 = 2,$$

$$bw(adj(A)) = 1 + 2 \cdot b2 = 13 \geq D+1 = 4, \quad w(G) = 13.$$

Natürlich ist diese Nummerierung nicht günstig gewesen. Eine Umnummerierung ist angebracht. Mit dem Permutationsvektor $p = (1, 2, 3, 5, 7, 8, 4, 6)$ erhält man den Graphen und die zugehörige Adjazenzmatrix



Die Bandbreiten sind wieder $bw(adj(A)) = w(G) = 9$, $b2(adj(A)) = 4$.

Gleichzeitig erkennt man die Stufenstruktur der Nummerierung.

1. Stufe L_1 , Wurzel	1	$ L_1 = 1$
2. Stufe L_2	2,3,4	$ L_2 = 3$
3. Stufe L_3	5,6,7	$ L_3 = 3$
4. Stufe L_4	8	$ L_4 = 1$

Tab. 1.1

Stufenstruktur L_k , $k = 1, 2, 3, 4$,
mit Wurzel 1

Damit ist gemäß [65] Formel (1.6) $b2 \leq 5$. Aber die Knotennummerndifferenz von verbundenen Knoten in aufeinanderfolgenden Stufen beträgt 4. Somit ist eine weitere Bandbreitenreduktion nicht möglich.

Es ist nun die Aufgabe, eine Nummerierung der Knoten zu suchen, bei der die Bandbreite w möglichst klein ist.

Der GPS läuft grob in 3 Schritten ab.

(A) Stufenstruktur und Bestimmung eines Pseudodurchmessers

Im diesem Schritt zerlegt man den Graphen in Stufen oder Schichten.

$L = (L_1, L_2, \dots, L_r)$ heißt Stufenstruktur oder Schichtung (engl. level structure) des Graphen (G, \sim) , wenn das Folgende gilt:

- (1) Die Mengen $L_i, i = 1(1)r$, (Stufen, Schichten) sind nichtleere disjunkte Teilmengen von $V(G)$. Ihre Vereinigung ist $V(G)$.
- (2) Für $g \in L_i, h \in L_j$ und $g \sim h$ ist $|i - j| \leq 1$.

r heißt die Tiefe der Schichtung. Ihre Breite k ist die Anzahl der Elemente der Schicht mit den meisten Elementen (Knoten).

Abbildung 1.1 zeigt, wie man einen Graphen in Schichten zerlegen kann.

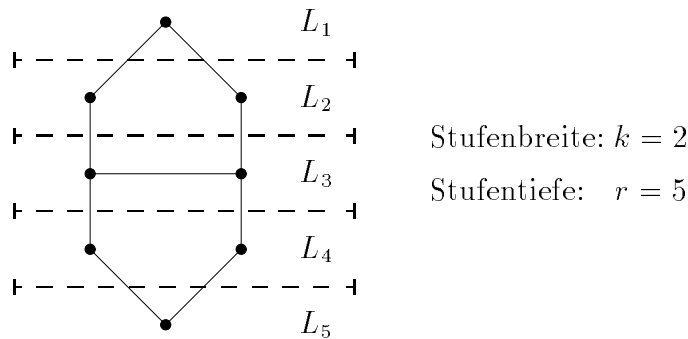


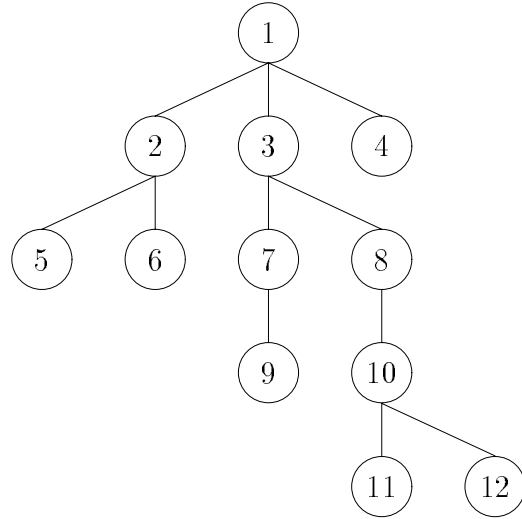
Abb. 1.1 Zerlegung eines Graphen in Schichten

Stufenstrukturen mit geringer Breite werden gewöhnlich unter denen mit größter Tiefe gesucht. Das heißt, wenn mehr Stufen vorhanden sind, entfallen im Schnitt weniger Knoten auf die einzelnen Stufen.

Es wäre ideal, Stufenstrukturen zu erzeugen, zu deren Wurzel $u \in V(G)$ ein Knoten v existiert mit $d(u, v) = \text{diam}(G)$. Das Problem ist zwar lösbar, meist wird jedoch ein approximativer Algorithmus dazu genommen. Dieser berechnet einen Pseudodurchmesser, der dem wahren Durchmesser $\text{diam}(G)$ nahe kommt.

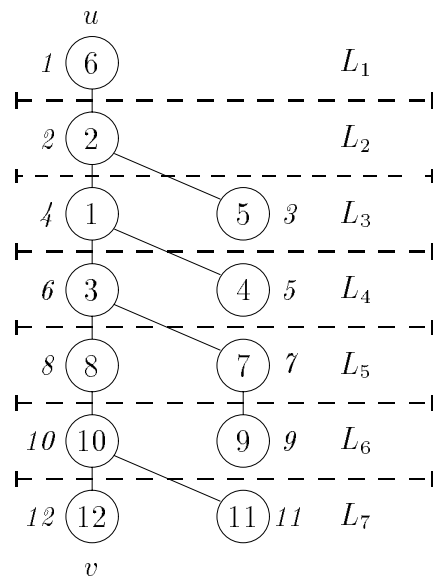
Für eine große Klasse von Graphen, einschließlich Bäume, ist der berechnete Pseudodurchmesser gleich dem wahren Durchmesser.

Beispiel 1.4 Der zusammenhängende Graph G mit 12 Knoten sei



Die Struktur entspricht keinem FEM-Netz.

Der Graph hat den Durchmesser $\text{diam}(G) = 6$. Nimmt man als Wurzel u den Knoten 6 (auch Knoten 5 möglich) und betrachtet dann das Kugelmodell, so ergibt sich die Stufenstruktur mit der Breite 2 und Tiefe 7.



Es ist weiterhin $D = 3$, $2 \leq b_2 \leq 3$, $5 \leq bw \leq 7$.

Die Umnummerierung der Knoten ist ebenfalls eingetragen. Sie entspricht dem Permutationsvektor $p = (4, 2, 6, 5, 3, 1, 7, 8, 9, 10, 11, 12)$.

Die bandbreitenreduzierte Matrix hat die Bandbreite $bw = 5$ und die Gestalt

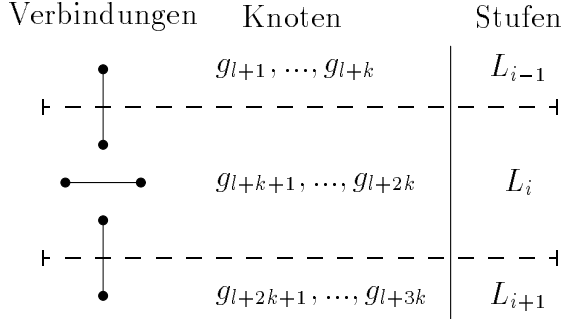
$$\begin{pmatrix} 1 & 1 & & & & & & & & & \\ 1 & 1 & 1 & 1 & & & & & & & \\ & 1 & 1 & & & & & & & & \\ & & 1 & & 1 & 1 & 1 & & & & \\ & & & 1 & 1 & & & & & & \\ & & & & 1 & 1 & 1 & & & & \\ & & & & & 1 & 1 & & 1 & & \\ & & & & & & 1 & & & 1 & \\ & & & & & & & 1 & & & 1 \\ & & & & & & & & 1 & 1 & 1 \\ & & & & & & & & & 1 & 1 \\ & & & & & & & & & & 1 \end{pmatrix}$$

Dass man die Stufenbreite klein machen sollte, unterstreicht auch der folgende Satz.

Satz 1.1 Es sei $L = (L_1, L_2, \dots, L_r)$ eine Stufenstruktur von (G, \sim) mit der Breite k . Dann gibt es eine Nummerierung der Knoten, so dass die Bandbreite des Graphen bezüglich dieser Nummerierung kleiner gleich $4k - 1$ ist.

Beweis

Man nummeriere erst die Elemente von L_1 , dann die Elemente von L_2 usw. Damit erhält man folgenden Ausschnitt aus der Stufenstruktur.



Für $g_\mu \sim g_\nu$, $g_\mu \in L_i$ und $g_\nu \in L_j$ folgen $|i - j| \leq 1$, $|\mu - \nu| \leq 2k - 1$ und damit $w(G) < 4k - 1$. □

Satz 1.2 Zu jedem Knoten g eines endlichen zusammenhängenden Graphen gibt es genau eine Stufenstruktur $R(g) = (L_1, L_2, \dots, L_r)$ folgender Art.

- (1) $L_1 = \{g\}$.
- (2) Zu jedem $h \in L_i$ mit $i > 1$ gibt es ein $f \in L_{i-1}$ mit $f \sim h$.

Diese Stufenstruktur heißt Schichtung mit der Wurzel g .

Beweis (konstruktiv) □

Im Folgenden sei der Graph (G, \sim) endlich und zusammenhängend. Gegebenenfalls kann man mehrere Zusammenhangskomponenten nacheinander nummerieren.

Algorithmus zur Bestimmung der Endknoten eines Pseudodurchmessers

- (A1) Man wähle einen Knoten $g \in V(G)$ mit minimalem Grad.
- (A2) Man erzeuge die Schichtung $R(g) = (L_1, L_2, \dots, L_r)$.
 Die Elemente der letzten Schicht L_r seien f_j , $j = 1(1)l$.
 Sie seien bez. ihres Grades geordnet, d. h. $d(f_j) \leq d(f_{j+1})$.
- (A3) Man setze $j = 1$.
- (A4) Man berechne $R(f_j) = (M_1, M_2, \dots, M_s)$ und setze m_j gleich der Breite der Schichtung $R(f_j)$, also $m_j = \max_{i=1(1)s} |M_i|$.
 Wenn $s > r$ ist, setze man $g = f_j$ und kehre nach (A2) zurück.
- (A5) Wenn $j < l$ ist, erhöhe man j um 1 und wiederhole (A4).
- (A6) Damit hat man in jedem Fall für den Knoten g die größte Strukturtiefe.
 Man wähle nun $j \in \{1, 2, \dots, l\}$ so, dass m_j minimal ist.
 Dann setze man $h = f_j$.

Dieser erste Teil bestimmt also zwei Knoten g und h als Endknoten eines Pseudodurchmessers und die zugehörigen Schichtungen

$$R(g) = (L_1, L_2, \dots, L_r), \quad L_1 = \{g\}, \quad \bigcup_{j=1}^r L_j = V(G), \quad L_j \text{ disjunkt},$$

$$R(h) = (M_1, M_2, \dots, M_s), \quad M_1 = \{h\}, \quad \bigcup_{j=1}^s M_j = V(G), \quad M_j \text{ disjunkt}.$$

Wir zeigen noch einige Eigenschaften der entstandenen Schichtungen $R(g)$ und $R(h)$.

Satz 1.3 Es gelten für die Schichtungen $R(g)$ und $R(h)$ gelten die Aussagen

- (1) $h \in L_r$,
- (2) $s \leq r$,
- (3) $M_i \subset \bigcup_{j=1}^i L_{r+1-j} = B_i$, $i = 1, 2, \dots, s$,
- (4) $s = r$,
- (5) $g \in M_r$,

$$(6) \quad L_i \subset \bigcup_{j=1}^i M_{r+1-j} = A_i, \quad i = 1, 2, \dots, r.$$

Beweis

(1) Diese Aussage folgt aus der Konstruktion (Schritte (A2) und (A6)) $h = k_j \in L_r$.

(2) $s \leq r$ folgt aus Schritt (A4) des Algorithmus.

(3) Für $i > s$ sei nun M_i die leere Menge.

Wir zeigen die Aussage durch vollständige Induktion. Wegen (1) ergibt sich der Induktionsanfang $M_1 = \{h\} \subset L_r$. Es folgt der Schluss von i auf $i + 1$. Alle Knoten aus M_{i+1} sind mit Knoten aus M_i verbunden (vergl. Satz 1.2 (2)). Weil die Elemente von L_μ nur mit Elementen in $L_{\mu-1}$, L_μ und $L_{\mu+1}$ verbunden sind und weil M_i eine Teilmenge von B_i ist, ist $M_{i+1} \subset B_{i+1}$.

(4) Aus (2) haben wir schon $s \leq r$. Aber s kann nicht kleiner als r sein, denn sonst hätte man mit $s = r - \alpha$, $\alpha \geq 1$,

$$M_s \subset \bigcup_{j=1}^s L_{r+1-j} = L_r \cup L_{r-1} \cup \dots \cup L_{\alpha+1} = B_s$$

einen Widerspruch wegen

$$V(G) = \bigcup_{j=1}^s M_j \subset L_r \cup L_{r-1} \cup \dots \cup L_{\alpha+1} \neq \bigcup_{j=1}^r L_j = V(G).$$

(5) Auf Grund von (3) und (4) folgen

$$\begin{aligned} M_1 &\subset L_r = B_1, \\ M_2 &\subset L_r \cup L_{r-1} = B_2, \\ &\dots \\ M_{r-1} &\subset L_r \cup L_{r-1} \cup \dots \cup L_2 = B_{r-1}, \\ M_r &\subset L_r \cup L_{r-1} \cup \dots \cup L_1 = B_r, \\ M_j &\subset B_j, \quad B_1 \subset B_2 \subset \dots \subset B_r, \\ \bigcup_{j=1}^{r-1} M_j &\subset \bigcup_{j=1}^{r-1} B_j = B_{r-1} = V(G) \setminus L_1, \quad V(G) = \bigcup_{j=1}^r M_j, \\ L_1 &\subset M_r, \\ g &\in M_r. \end{aligned}$$

(6) Der Nachweis kann wegen (5) analog zum Beweis von (3) geführt werden.

□

Die Aussagen des Satzes 1.3 müssen durch einige Erfahrungen ergänzt werden. In den meisten praktischen Anwendungen zeigt sich Folgendes.

- (6) Die Tiefe r der beiden Schichtungen $R(g)$ und $R(h)$ ist entweder die maximale Tiefe, die bei einer Schichtung von (G, \sim) erreicht werden kann, also der Durchmesser, oder eine gute Näherung davon.
- (7) Die Mengen $L_j \cap M_{r+1-j}$, $j = 1(1)r$, enthalten in vielen Fällen die meisten Elemente von $L_j \cup M_{r+1-j}$. Die beiden Schichtungen sind also sehr ähnlich, selten allerdings identisch.
- (8) Der Rücksprung von Schritt (A4) des Algorithmus zu Schritt (A2) kommt nur bei wenigen Beispielen überhaupt vor, dann aber nicht häufiger als einmal bei einem Graphen. Diese Beobachtung ist selbstverständlich von großer Bedeutung für die Rechenzeit.

Beispiel 1.5 Wir betrachten einen zusammenhängenden Graphen G mit 6 Knoten und $V(G) = \{v, t, x, r, s, y\}$.

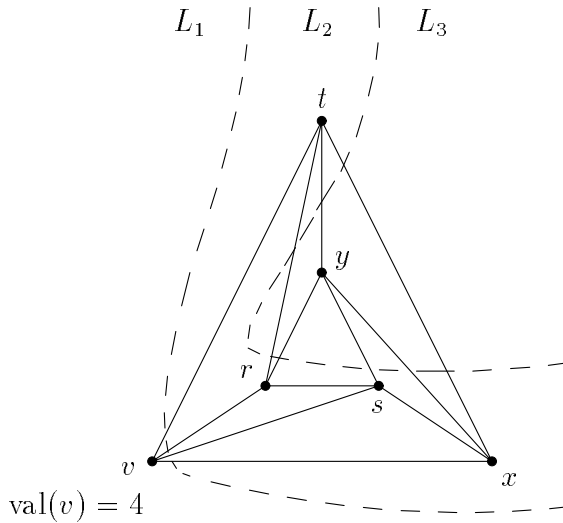


Abb. 1.2
Graph mit logischer Stufenstruktur

Alle Knoten haben den Grad 4 und der Durchmesser ist 2.

Wir konstruieren die Stufenstruktur mit dem Startknoten v . Damit ergibt sich auf Grund der Verbindungen der Knoten die Schichtung

$$R(v) = (L_1, L_2, L_3) = (\{v\}, \{t, r, s, x\}, \{y\})$$

mit der Tiefe 3 und Breite 4.

Da in der tiefsten Stufe nur der Knoten y mit dem Grad 4 ist, kontrollieren wir die zugehörige Schichtung

$$R(y) = (M_1, M_2, M_3) = (\{y\}, \{t, r, s, x\}, \{v\}),$$

die sehr ähnlich zu $R(v)$ ist. Die Tiefe hat sich dabei nicht vergrößert, so dass die Knoten v und y auch den Durchmesser 2 erzeugen.

Ein Hinweis auf ein nichtkorrektes Vorgehen an dieser Stelle.
Nicht verwendet werden kann die Stufenstruktur

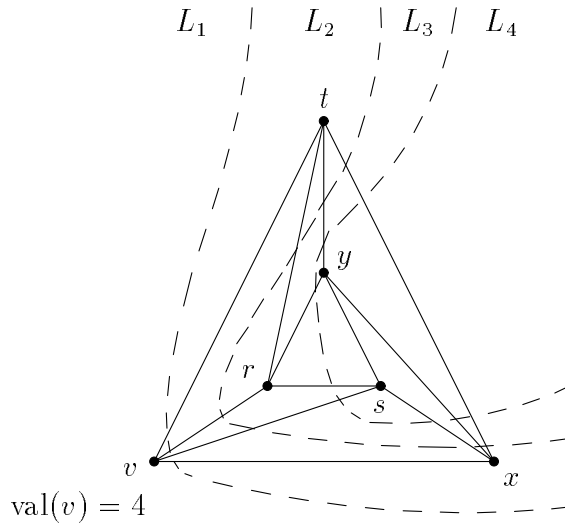


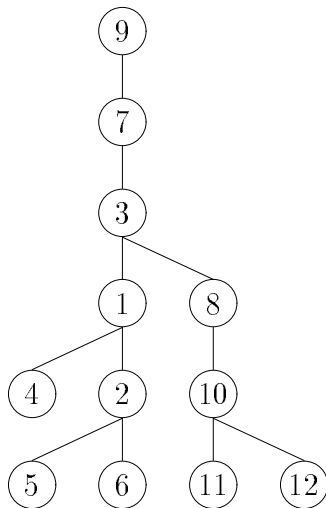
Abb. 1.3

Graph mit unzulässiger Stufenstruktur

mit $R(v) = (L_1, L_2, L_3, L_4) = (\{v\}, \{t, x\}, \{r\}, \{y, s\})$ bei der Tiefe 4 und Breite 2. Die Verbindungen $v \sim r$ und $v \sim s$ verlangen, dass v sowie r, s in benachbarten Stufen liegen. Würde man trotzdem noch die Schichtung für den Knoten y der tiefsten Stufen gemäß $R(y) = (M_1, M_2, M_3, M_4, M_5, M_6) = (\{y\}, \{t\}, \{v\}, \{x\}, \{s\}, \{r\})$ durchführen, so träte wegen der Tiefe $6 > 4$ dann der Fall des Rücksprungs zum Schritt (A2) im Algorithmus auf.

Beispiel 1.6 Kehren wir zum Graphen im Beispiel 1.4 zurück.

Würden wir diesen wie ein Kugelmodell am Knoten 9 mit dem kleinsten Grad 1 aufhängen, so ergibt sich die Schichtung $R(9) = (L_1, L_2, \dots, L_6)$ mit $L_1 = \{9\}, L_2 = \{7\}, \dots, L_6 = \{5, 6, 11, 12\}$.



Alle Knoten in der tiefsten Stufe haben den Grad 1, so dass dann im Algorithmus (A) die Berechnung von $R(5)$ eine tiefere Stufenstruktur liefert, nämlich

$$R(5) = (M_1, M_2, \dots, M_6, M_7) \text{ mit } M_1 = \{5\} \text{ und } M_7 = \{11, 12\}.$$

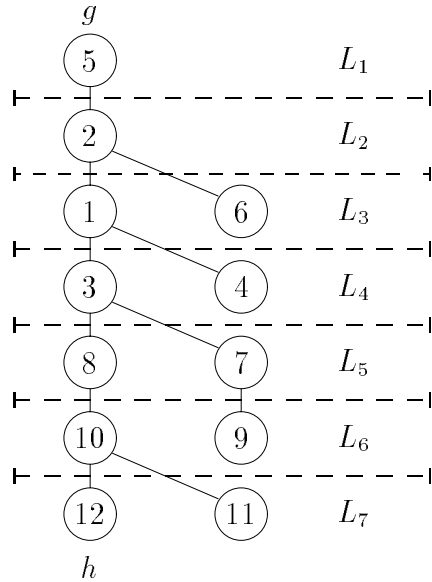
Die Wiederholung ab Schritt (A2) mit der Wurzel 5 liefert nun keine tieferen Stufenstrukturen für die "Endknoten" 11 und 12 mehr. Damit erhält man

$R(5)$	$R(12)$
$L_1 = \{5\}$	$M_1 = M_{8-7} = \{12\}$
$L_2 = \{2\}$	$M_2 = M_{8-6} = \{10, 11\}$
$L_3 = \{1, 6\}$	$M_3 = M_{8-5} = \{8\}$
$L_4 = \{3, 4\}$	$M_4 = M_{8-4} = \{3\}$
$L_5 = \{7, 8\}$	$M_5 = M_{8-3} = \{1, 7\}$
$L_6 = \{9, 10\}$	$M_6 = M_{8-2} = \{2, 4, 9\}$
$L_7 = \{11, 12\}$	$M_7 = M_{8-1} = \{5, 6\}$

Tab. 1.2

Schichtungen nach Algorithmus (A)

Die Schichtungen $R(5)$ und $R(12)$ haben die gleichen Tiefen 7, aber unterschiedliche Breiten 2 bzw. 3. Der Durchmesser ist 6, bestimmt durch die Endknoten 5 und 12.



(B) Bestimmung von S und Minimierung der Stufenbreite

Da die Schichtungen $R(g)$, $L_1 = \{g\}$, $h \in L_r$, und $R(h)$, $M_1 = \{h\}$, $g \in M_r$, sehr ähnlich sind, wird im nächsten Teil des Algorithmus aus den beiden Schichtungen eine neue Schichtung

$$S = (S_1, S_2, \dots, S_r)$$

mit gleicher Tiefe und möglichst kleiner Breite konstruiert. Jeder Knoten $f \in V(G)$ wird dabei nach einer der beiden folgenden Regeln in eine Schicht S_i eingeordnet.

Regel 1: Falls $f \in L_i$ ist, so sei $f \in S_i$.

Regel 2: Falls $f \in M_{r+1-i}$ ist, so sei $f \in S_i$.

Für die Elemente von $L_i \cap M_{r+1-i}$ ergeben beide Regeln das gleiche Ergebnis. Darum ist jedenfalls

$$L_i \cap M_{r+1-i} \subset S_i.$$

Die Menge

$$V(G') = V(G) \setminus \bigcup_{i=1}^r L_i \cap M_{r+1-i}$$

des Restgraphen G' zerfällt in l Zusammenhangskomponenten G'_1, G'_2, \dots, G'_l .

Leider ist es nicht möglich, für die einzelnen Elemente von $V(G)$ unabhängig voneinander eine der Regeln 1 und 2 anzuwenden. Ein solches Vorgehen ergibt im Allgemeinen keine Schichtung S . Es gilt aber

Satz 1.4 Wenn in jeder Zusammenhangskomponente von G' konstant die gleiche Regel (entweder immer 1 oder immer 2) angewandt wird, ist S eine Schichtung.

Beweis

Die Anwendung der gleichen Regel in allen G'_j bedeutet die Verwendung derselben für alle $h \in V(G)$, so dass sich die Schichtung von $R(g)$ bzw. von $R(h)$ auf S überträgt. \square

Weiterhin definieren wir den Begriff des assoziierten Stufenpaares.

Jedem Knoten $f \in V(G)$ ordnen wir das assoziierte Stufenpaar (i, j) (geordnetes Paar) zu, wobei

- a) i der Index der Stufe in $R(g)$ ist, in der f enthalten ist, und
- b) $r + 1 - j$ der Index der Stufe in $R(h)$ ist, in der f enthalten ist.

Damit ist das Paar (i, j) assoziiert mit dem Knoten f genau dann, wenn $f \in L_i \cap M_{r+1-j}$ ist.

Wegen $L_1 = \{g\}$, $h \in L_r$, $M_1 = \{h\}$, $g \in M_r$ erhält man sofort, dass

- $(1, 1)$ assoziiert mit g ist, da $g \in L_1 \cap M_r$,
- (r, r) assoziiert mit h ist, da $h \in L_r \cap M_1$.

Die Schichtung $R(g)$ hat die Breite $k_1 = \max_{i=1(1)r} |L_i|$, die Schichtung $R(h)$ die Breite

$$k_2 = \max_{i=1(1)r} |M_i|.$$

Dieser zweite Teil des Algorithmus zu Ermittlung von $S = (S_1, S_2, \dots, S_r)$ besteht aus drei Einzelschritten.

Dabei geben wir 2 Darstellungsmöglichkeiten (1) bzw. (2) an.

- (B1) (1) Man setze $S_i = L_i \cap M_{r+1-i}$, $i = 1(1)r$, und bestimme $V(G')$. Wenn $V(G')$ die leere Menge ist, dann ist dieser Teil (B) fertig.
 (2) Man nimmt der Reihe nach alle Knoten $f \in V(G)$. Ist das mit dem Knoten assoziierte Paar der Form (i, i) , so wird dieser Knoten f der Stufe S_i zugeordnet. Der Knoten f und alle mit ihm inzidenten (verknüpften) Kanten werden aus G entfernt.
 Wenn $V(G)$ leer ist, dann Ende von (B).
- (B2) (1) Man zerlege G' in Zusammenhangskomponenten G'_1, G'_2, \dots, G'_l , $l \geq 1$.
 Man sortiere die G'_j so, dass stets $|G'_j| \geq |G'_{j+1}|$, $j = 1(1)l - 1$, gilt.
 (2) Der Restgraph G' besteht aus einer oder mehreren disjunkten zusammenhängenden Komponenten G'_1, G'_2, \dots, G'_l , $l \geq 1$, die mit wachsendem Index abnehmende Knotenzahl haben sollen.
- (B3) Für jede Komponente G'_j , $j = 1(1)l$, führe man die folgenden Schritte aus.
- (1.1) Man erweitere alle S_i zu \tilde{S}_i , $i = 1(1)r$, durch Einordnung der Knoten aus G'_j nach Regel 1.
 Man erweitere alle S_i zu \bar{S}_i , $i = 1(1)r$, durch Einordnung der Knoten aus G'_j nach Regel 2.
- (1.2) Man berechne
- $$k_3 = \max(|\tilde{S}_i| : i = 1(1)r \text{ mit } \tilde{S}_i \neq S_i)$$
- und
- $$k_4 = \max(|\bar{S}_i| : i = 1(1)r \text{ mit } \bar{S}_i \neq S_i).$$
- (1.3) Falls $k_3 < k_4$ oder $k_3 = k_4$ und $k_1 \leq k_2$ sind, setze man $S_i = \tilde{S}_i$,
 sonst $S_i = \bar{S}_i$, $i = 1(1)r$.
- (2.1) Man berechne den Breitenvektor $s = (s_1, s_2, \dots, s_r)$ mit $s_i = |S_i|$.
 Man bestimme die Breitenvektoren $(\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_r)$ bzw. $(\bar{s}_1, \bar{s}_2, \dots, \bar{s}_r)$ mit
- $$\tilde{s}_i = s_i + \text{“Anzahl der Knoten, die in } S_i \text{ plaziert werden, wenn das 1. Element des Paares benutzt wurde, also gleich } i \text{ ist (} f \in L_i \text{)“},$$
- $$\bar{s}_i = s_i + \text{“Anzahl der Knoten, die in } S_i \text{ plaziert werden, wenn das 2. Element des Paares benutzt wurde, also gleich } i \text{ ist (} f \in M_{r+1-i} \text{)“}.$$
- (2.2) Man berechne
- $$k_3 = \max_{i=1(1)r} (\tilde{s}_i : \tilde{s}_i - s_i > 0)$$
- und
- $$k_4 = \max_{i=1(1)r} (\bar{s}_i : \bar{s}_i - s_i > 0).$$

- (2.3) – Falls $k_3 < k_4$ ist, dann ordne man alle Knoten der Komponente in die von den 1. Elementen der assoziierten Paare angezeigten Stufen von S ein.
 – Falls $k_3 > k_4$ ist, dann ordne man alle Knoten der Komponente in die von den 2. Elementen der assoziierten Paare angezeigten Stufen von S ein.
 – Falls $k_3 = k_4$ ist, dann benutze man diejenigen Elemente der assoziierten Paare, die von der Stufenstruktur mit der geringeren Breite herrühren. Sind die Breiten k_1 und k_2 gleich, dann benutze man die 1. Elemente.

Damit ist jeder Knoten von G in eine Stufe von S eingeordnet und die Berechnung der Schichtung S abgeschlossen.

Dieser zweite Teil des Algorithmus beansprucht einen großen Teil der Rechenzeit. Das gilt insbesondere, wenn $V(G')$ viele Komponenten besitzt.

Beispiel 1.7 In Abbildung 1.4 sind für einen Graphen die Schichtungen $R(g)$, $R(h)$ und S aufgezeichnet, wobei das Ergebnis S natürlich noch zu begründen ist.

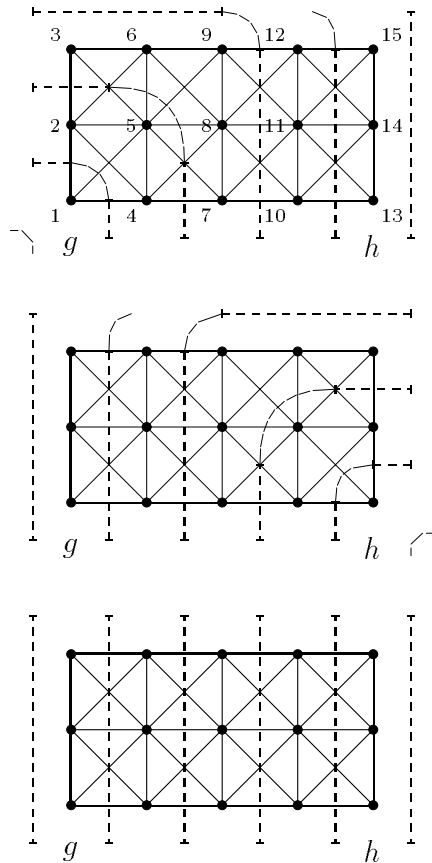
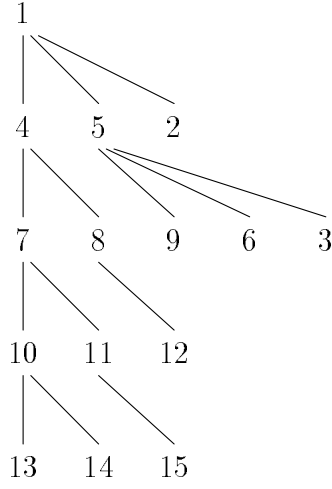
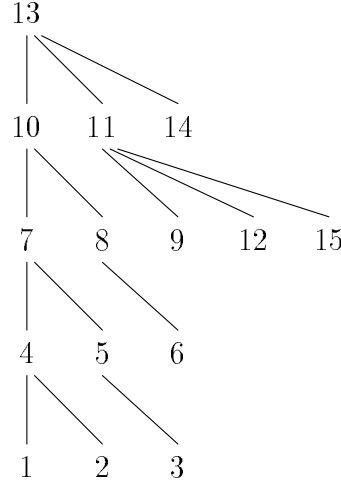


Abb. 1.4 Graph mit Schichtungen $R(g)$, $R(h)$ und S

Schichtung $R(g)$, $g = 1$ 

Grad: 3 5 3

Schichtung $R(h)$, $h = 13$ 

Grad: 3 5 3

$R(g)$	$R(h)$	Init. S	$ S_i $
$L_1 = \{1\}$	$M_1 = M_{6-5} = \{13\}$	$S_1 = \{1\}$	1
$L_2 = \{4, 5, 2\}$	$M_2 = M_{6-4} = \{10, 11, 14\}$	$S_2 = \{4, 5\}$	2
$L_3 = \{7, 8, 9, 6, 3\}$	$M_3 = M_{6-3} = \{7, 8, 9, 12, 15\}$	$S_3 = \{7, 8, 9\}$	3
$L_4 = \{10, 11, 12\}$	$M_4 = M_{6-2} = \{4, 5, 6\}$	$S_4 = \{10, 11\}$	2
$L_5 = \{13, 14, 15\}$	$M_5 = M_{6-1} = \{1, 2, 3\}$	$S_5 = \{13\}$	1

Tab. 1.3 Schichtungen nach Algorithmus (A) mit Initialisierung von S

$V(G')$ besteht hier aus zwei Komponenten mit je drei Knoten.

$$V(G') = \{2, 3, 6, 12, 14, 15\} = \{\{2, 3, 6\}, \{12, 14, 15\}\} = \{G'_1, G'_2\}, \quad |G'_j| = 3.$$

Betrachten wir den Schritt (B3) des Algorithmus. Für die linke Komponente wird Regel 2, für die rechte Komponente Regel 1 angewandt.

Schema:

$$j = 1, \quad G'_1 = \{2, 3, 6\}$$

S_i	1	4,5	7,8,9	10,11	13	
\tilde{S}_i	1	2,4,5	3,7,8,9,6	10,11	13	Regel 1, $k_3 = 5$
\bar{S}_i	1,2,3	4,5,6	7,8,9	10,11	13	Regel 2, $k_4 = 3$
<hr/>						
$k_3 > k_4$						
$S_i = \bar{S}_i$	also Anwendung von Regel 2					
S_i	1,2,3	4,5,6	7,8,9	10,11	13	

$$j = 2, \quad G'_2 = \{12, 14, 15\}$$

S_i	1,2,3 4,5,6 7,8,9 10,11 13	
\tilde{S}_i	1,2,3 4,5,6 7,8,9 10,11,12 13,14,15	Regel 1, $k_3 = 3$
\bar{S}_i	1,2,3 4,5,6 7,8,9,12,15 10,11,14 13	Regel 2, $k_4 = 5$
<hr/>		
$k_3 < k_4$		
$S_i = \tilde{S}_i$	also Anwendung von Regel 1	
S_i	1,2,3 4,5,6 7,8,9 10,11,12 13,14,15	

So ergibt sich die vertikale Schichtung mit der optimalen Breite 3.

Dies entspricht auch der Vorgehensweise bei der Nummerierung eines rechtwinkligen Gitternetzes, wo der Durchlauf in der Koordinatenrichtung zuerst erfolgt, in der die kleinere Anzahl von Gitterpunkten liegt.

Beispiel 1.8 Wir knüpfen an das Beispiel 1.6 an mit

$R(5), k_1 = 2$	$R(12), k_2 = 3$	Init. S
$L_1 = \{5\}$	$M_1 = M_{8-7} = \{12\}$	$S_1 = \{5\}$
$L_2 = \{2\}$	$M_2 = M_{8-6} = \{10, 11\}$	$S_2 = \{2\}$
$L_3 = \{1, 6\}$	$M_3 = M_{8-5} = \{8\}$	$S_3 = \{1\}$
$L_4 = \{3, 4\}$	$M_4 = M_{8-4} = \{3\}$	$S_4 = \{3\}$
$L_5 = \{7, 8\}$	$M_5 = M_{8-3} = \{1, 7\}$	$S_5 = \{8\}$
$L_6 = \{9, 10\}$	$M_6 = M_{8-2} = \{2, 4, 9\}$	$S_6 = \{10\}$
$L_7 = \{11, 12\}$	$M_7 = M_{8-1} = \{5, 6\}$	$S_7 = \{12\}$

Tab. 1.4 Schichtungen nach Algorithmus (A) mit Initialisierung von S

Wir untersuchen nun die 2. Darstellungsmöglichkeit im Algorithmus (B) unter Verwendung der assoziierten Paare für die Knoten.

Knoten	assoz. Paare (i, j)	Zuordnung zu S_i
1	(3,3)	$1 \in L_3 \cap M_{8-3} \rightarrow 1 \in S_3$
2	(2,2)	$2 \in L_2 \cap M_{8-2} \rightarrow 2 \in S_2$
3	(4,4)	$3 \in L_4 \cap M_{8-4} \rightarrow 3 \in S_4$
4	(4,2)	
5	(1,1)	$5 \in L_1 \cap M_{8-1} \rightarrow 5 \in S_1$
6	(3,1)	
7	(5,3)	
8	(5,5)	$8 \in L_5 \cap M_{8-5} \rightarrow 8 \in S_5$
9	(6,2)	
10	(6,6)	$10 \in L_6 \cap M_{8-6} \rightarrow 10 \in S_6$
11	(7,6)	
12	(7,7)	$12 \in L_7 \cap M_{8-7} \rightarrow 12 \in S_7$

Tab. 1.5 Knoten 1–12 mit assoz. Paaren und Zuordnung zu den Schichten

Alle Breiten $|S_i|$ sind 1.

Der Graph G zerfällt nach Streichung der Knoten 1,2,3,5,8,10,12 und inzidenten Kanten in 4 Untergraphen (Zusammenhangskomponenten).

$$V(G') = \{4, 6, 7, 9, 11\} = \{\{7, 9\}, \{4\}, \{6\}, \{11\}\} = \{G'_1, G'_2, G'_3, G'_4\},$$

die mit abnehmender Knotenzahl sortiert sind.

Für die 4 Untergraphen betrachten wir zu S_i die Erweiterungen \tilde{S}_i und \bar{S}_i .

Schema:

$$j = 1, G'_1 = \{7, 9\}, 7 \rightarrow (5, 3), 9 \rightarrow (6, 2)$$

S_i	5	2	1	3	8	10	12	
\tilde{S}_i	5	2	1	3	8,7	10,9	12	$k_3 = 1$
\bar{S}_i	5	2,9	1,7	3	8	10	12	$k_4 = 1$
<hr/>								
$k_3 = k_4 = 1$								
$S_i = \tilde{S}_i$	also Anwendung von Regel 1							
S_i	5	2	1	3	8,7	10,9	12	
$s_i = S_i $	1	1	1	1	2	2	1	

$$j = 2, G'_2 = \{4\}, 4 \rightarrow (4, 2)$$

S_i	5	2	1	3	8,7	10,9	12	
\tilde{S}_i	5	2	1	3,4	8,7	10,9	12	$k_3 = 1$
\bar{S}_i	5	2,4	1,7	3	8,7	10,9	12	$k_4 = 1$
<hr/>								
$k_3 = k_4 = 1$								
$S_i = \tilde{S}_i$	also Anwendung von Regel 1							
S_i	5	2	1	3,4	8,7	10,9	12	
$s_i = S_i $	1	1	1	2	2	2	1	

$$j = 3, G'_3 = \{6\}, 6 \rightarrow (3, 1)$$

S_i	5	2	1	3,4	8,7	10,9	12	
\tilde{S}_i	5	2	1,6	3,4	8,7	10,9	12	$k_3 = 1$
\bar{S}_i	5,6	2	1	3,4	8,7	10,9	12	$k_4 = 1$
<hr/>								
$k_3 = k_4 = 1$								
$S_i = \tilde{S}_i$	also Anwendung von Regel 1							
S_i	5	2	1,6	3,4	8,7	10,9	12	
$s_i = S_i $	1	1	2	2	2	2	1	

$j = 4$, $G'_4 = \{11\}$, $11 \rightarrow (7, 6)$

S_i	5	2	1,6	3,4	8,7	10,9	12	
\tilde{S}_i	5	2	1,6	3,4	8,7	10,9	12,11	$k_3 = 1$
\bar{S}_i	5	2,11	1,6	3,4	8,7	10,9	12	$k_4 = 1$
<hr/>								
$k_3 = k_4 = 1$								
$S_i = \tilde{S}_i$	also Anwendung von Regel 1							
S_i	5	2	1,6	3,4	8,7	10,9	12,11	
$s_i = S_i $	1	1	2	2	2	2	2	

Es wird also immer die Regel 1 angewendet, d. h.

$$S = (S_1, S_2, \dots, S_7) = R(5) = (L_1, L_2, \dots, L_7).$$

(C) Umnummerierung der Knoten

Ausgehend von der Schichtung $S = (S_1, S_2, \dots, S_l, \dots, S_r)$ kann nun im dritten und letzten Teil des Algorithmus der Graph nummeriert werden.

Variante (1)

Die Anfangsnummer sei 0.

(C1) Man setze $l = 1$. Man gehe zu (C4).

(C2) f durchlaufe alle Elemente von S_{l-1} in der Reihenfolge der Nummerierung. Für festes f nummeriere man alle noch nicht nummerierten $\tilde{f} \in S_l$, die mit f verbunden sind, so dass der Grad der \tilde{f} nicht fällt.

(C3) f durchlaufe alle bereits nummerierten Elemente von S_l in der Reihenfolge der Nummerierung. Für festes f nummeriere man alle noch nicht nummerierten $\tilde{f} \in S_l$, die mit f verbunden sind, so dass der Grad von \tilde{f} nicht fällt.

(C4) Wenn noch nicht alle Elemente von S_l nummeriert sind, suche man ein unnummeriertes Element aus S_l mit minimalem Grad. Man gebe ihm die **nächste Nummer** und kehre nach (C3) zurück.

(C5) Man erhöhe l um 1 und kehre für $l \leq r$ nach (C2) zurück.

Diese Variante lässt auch zu, dass in den Schichten S_l Knoten ohne Verbindung zu S_{l-1} auftreten bzw. mit anderen Knoten in S_l verbunden sind.

Berücksichtigt man die Besonderheit der Schichtung, dass $S_1 = L_1 = \{g\}$ und $M_1 = \{h\}$ sind, die bestehenden Verbindungen zwischen den Schichten sowie die Grade von g und h , ergibt sich eine zweite Darstellungsmöglichkeit von Teil (C).

Variante (2)

- (C1) Wenn der Grad von h kleiner als der von g ist, kehre man durch Setzen von S_l zu S_{r+1-l} , $l = 1, 2, \dots, r$, die Reihenfolge der Stufen um.
- (C2) Man setze für $g \in S_1 \subset V(G)$ die Anfangsnummer 1.
- (C3) Für $l = 2, 3, \dots, r$ führe man aus:
- (C3.1) Sei $f \in S_{l-1}$ (falls er existiert) derjenige Knoten, der die niedrigste Nummer hat und allen bereits nummerierten Knoten in S_{l-1} , die zu wenigstens einem noch unnummerierten Knoten aus S_l adjazent sind, in der Reihenfolge aufsteigender Grade.
Man gebe diesem unnummerierten Knoten aus S_l die nächste Nummer.
 - (C3.2) Man wiederhole (C3.1) jeweils für den nächsten Knoten aus S_{l-1} , bis alle Knoten aus S_l , die zu Knoten aus S_{l-1} adjazent sind, nummeriert wurden.

Mit diesem Schritt ist auch die Nummerierung beendet.

Alte Nummerierung des Graphen: $1, 2, \dots, n$.

Neue Nummerierung des Graphen: $\pi_1, \pi_2, \dots, \pi_n$.

Die neue Nummerierung ist eine Permutation des Vektors $(1, 2, \dots, n)$.

Die Permutationsmatrix P für die Transformation $A' = P^T A P$ hat ihre Eins-Einträge an den Stellen (i, π_i) , $i = 1, 2, \dots, n$.

Das Ergebnis lässt sich in einigen Fällen noch durch einen iterativen Algorithmus von Rosen 1968 [61] verbessern. Über Erfahrungen mit diesem Algorithmus berichtet Lierz 1976 [62].

Alle Algorithmen zur Bandbreitenreduktion kommen hauptsächlich für nicht zu große schwach besetzte Matrizen mit unregelmäßigem Muster in Frage. Sie treten hauptsächlich bei Finite-Element-Ansätzen mit komplizierter Geometrie auf. Eine typische Größenordnung wäre etwa $n = 1000$, $bw = 50$.

1.1.2 Programmierläuterungen zu GPS

Das FORTRAN-Programm zu GPS ist aus [29].

Der dortige Algorithmus mit den Schritten A-O ist leicht modifiziert worden.

Bezogen auf seine drei Hauptteile ist die Zuordnung dieser Schritte zur neuen Einteilung wie folgt.

A	A1	G	B1,B2	K	C1 (Variante (1))
B	A2	H,I,J	B3	L	C2
C	A3			M	C3
D	A4			N	C4
E	A5			O	C5
F	A6				

```

C -----
C   HP : BANDRED1.FOR  DOS-Version
C   UP : REDUCE
C -----
C   HAUPTPROGRAMM ZUR BANDBREITENREDUKTION EINER SYMMETRISCHEN
C   SCHWACH BESETZTEN MATRIX NACH DEM VERFAHREN VON GIBBS, POOLE
C   UND STOCKMEYER
C   TH.MEIS, U.MARCOWITZ : NUM. BEHANDLUNG PARTIELLER DIFFERENTIAL-
C   GLEICHUNGEN; SPRINGER BERLIN 1978, KAP. A5
C
C   DAS PROGRAMM IST AUSGELEGT FUER MAXIMAL
C       NMAX  = 10000 MATRIXZEILEN
C
C   EINGABE
C       DATEI  MIT  N, M, A(I)
C       N      = ANZAHL DER ZEILEN DER MATRIX
C               BEGRENZUNG DURCH NMAX (=650)
C       M      = ANZAHL DER VON NULL VERSCHIEDENEN MATRIXELEMENTE
C               OBERHALB DER HAUPTDIAGONALEN
C               BEGRENZUNG DURCH MMAX (=2048)
C       A(I), I=1(1)2*M
C               ENTHAELT DIE INDIZES DER NNE DER MATRIX OBER-
C               HALB DER HAUPTDIAGONALEN IN GEORDNETER REIHEN-
C               FOLGE  I1,J1, I2,J2, ...
C   AUSGABE
C       DATEI  MIT KOLD, KNEW, N, A(I)
C       KOLD   = BANDBREITE VOR DER BANDBREITENREDUKTION
C       KNEW   = BANDBREITE DANACH
C       N
C       A(I), I=1(1)N
C               PERMUTATIONSVEKTOR FUER ZEILEN UND SPALTEN

```

```

C -----
      PARAMETER(NMAX=650,MMA=2048)
      INTEGER  A(4096),VEC(650),IND(651,8)
      INTEGER  N,M,KOLD,KNEW,I
      CHARACTER*48  FNAME1,FNAME3,FNAME4
      COMMON A,VEC,IND
      WRITE(*,899)
899  FORMAT('  BANDBREITENREDUKTION EINER SYMM. MATRIX'/)
      WRITE(*,900)
900  FORMAT('  NAME DER EINGABEDATEI      (...MATR*.DAT) :  ')
      READ(*,'(A12)')  FNAME1
      OPEN(1,FILE=FNAME1,STATUS='OLD')
      WRITE(*,901)
901  FORMAT('  NAME DER 1. AUSGABEDATEI  (...PERM*.DAT) :  ')
      READ(*,'(A12)')  FNAME3
      OPEN(3,FILE=FNAME3,STATUS='UNKNOWN')
      WRITE(*,902)
902  FORMAT('  NAME DER 2. AUSGABEDATEI (...PERMC*.DAT) :  ')
      READ(*,'(A12)')  FNAME4
      OPEN(4,FILE=FNAME4,STATUS='UNKNOWN')
      READ(1,*)  N,M
      IF(N.LT.2 .OR. N.GT.NMAX)  STOP  'N ZU KLEIN/GROSS !!'
      IF(M.GT.MMA)  STOP  'M ZU GROSS !!'
      READ(1,*)  (A(I), I=1,2*M)
10  CONTINUE

C -----
C      AUFRUF DES UP  REDUCE
C -----
      CALL  REDUCE(N,M,KOLD,KNEW)
C -----
C      ERGEBNISSE IN DATEIN
C -----
      WRITE(3,5)  KOLD,KNEW
5  FORMAT(/1X,2I5)
      WRITE(3,6)  N
6  FORMAT(/1X,1I5)
      WRITE(3,7)  (A(I), I=1,N)
7  FORMAT((1X,10I5))
      WRITE(4,7)  (A(I), I=1,N)
      STOP  'S C H L U S S'
      END

C
C -----
C      UNTERPROGRAMME

```

```

C -----
C
C      SUBROUTINE REDUCE(N,M,KOLD,KNEW)
C -----
C      UNTERPROGRAMM ZUR BANDBREITENREDUKTION EINER SYMMETRISCHEN
C      SCHWACH BESETZTEN MATRIX NACH DEM VERFAHREN VON GIBBS, POOLE
C      UND STOCKMEYER
C      DIE DIMENSIONSGRENZEN KOENNEN GEAENDERT WERDEN, ABER DAS
C      PROGRAMM IST AUSGELEGT FUER MAXIMAL
C      NMAX  = 10000 MATRIXZEILEN
C
C      EINGABE
C      N      = ANZAHL DER ZEILEN DER MATRIX
C              BEGRENZUNG DURCH NMAX (=650)
C      M      = ANZAHL DER VON NULL VERSCHIEDENEN MATRIXELEMENTE
C              OBERHALB DER HAUPTDIAGONALEN
C              BEGRENZUNG DURCH MMAX (=2048)
C      A(I), I=1(1)2*M
C              ENTHAELT DIE INDIZES DER NNE DER MATRIX OBER-
C              HALB DER HAUPTDIAGONALEN IN GEORDNETER REIHEN-
C              FOLGE  I1,J1, I2,J2, ...
C      AUSGABE
C      KOLD   = BANDBREITE VOR DER BANDBREITENREDUKTION
C      KNEW   = BANDBREITE DANACH, WENN DIE EINGABEMATRIX A(I)
C              ENTSPRECHEND PERMUTIERT WIRD
C      A(I), I=1(1)N
C              PERMUTATIONSVEKTOR FUER ZEILEN UND SPALTEN
C              NMAX<=2*MMAX
C -----
C      A(2*MMAX)
C      VEC(NMAX), IND(NMAX+1,8), LIST(NMAX+1), GRAD(NMAX), NR(NMAX)
C -----
C
C      INTEGER M,N,KOLD,KNEW
C      INTEGER A(4096),VEC(650),
C      *      IND(651,8),LIST(651),GRAD(650),NR(650)
C      COMMON A,VEC,IND
C      EQUIVALENCE (LIST(1),IND(1,1)),(GRAD(1),IND(1,2)),
C      *      (NR(1),IND(1,3))
C      INTEGER NN,M2,N1,NUE,N1M,NUM,IS,OLD,NEW,
C      *      F,L,L1,L2,L10,I,J,K,III,KNP1,K1,K1N,K2,K2P1,
C      *      G,H,START,DEPTHF,DEPTHB,LEWVTH,GRDMIN,C,C2,
C      *      KAPPA,KAPPA1,KAPPA2,KAPPA3,KAPPA4,
C      *      IND1,IND2,INDJ2,INDJ5,INDJ6,INDI7,INDI8,VECJ
C      DATA C/10000/

```

```

      C2=2*C
C -----
C   ABSCHNITT 1
C -----
      M2=M+M
      KOLD=0
      KNEW=N
      DO 10 I=1,8
        DO 10 J=1,N
10    IND(J,I)=0
      IF(M.EQ.0) GOTO 680
      DO 15 I=1,M2,2
        J=IABS(A(I)-A(I+1))
        IF(J.GT.KOLD) KOLD=J
15  CONTINUE
      DO 20 I=1,M2
        K1=A(I)
20    IND(K1,7)=1
      NUM=1
      DO 30 I=1,N
        IF(IND(I,7).GT.0) GOTO 30
        NR(I)=NUM
        NUM=NUM+1
30  CONTINUE
C -----
C   ABSCHNITT 2 (NEUE DATENSTRUKTUR)
C -----
      DO 40 I=1,M2,2
        K1=A(I)
        K2=A(I+1)
        A(I)=K1+C*K2
40    A(I+1)=K2+C*K1
      CALL SSORT1(1,M2)
      J=1
      OLD=A(1)
      DO 70 I=2,M2
        NEW=A(I)
        IF(NEW.GT.OLD) J=J+1
        A(J)=NEW
70    OLD=NEW
      M2=J
      IND(1,2)=1
      J=1
      L10=A(1)/C
      DO 90 I=1,M2
        K=A(I)

```

```

      L1=K/C
      L2=K-L1*C
      A(I)=L2
      IF(L1.EQ.L10) GOTO 90
      L10=L1
      J=J+1
      IND(J,2)=I
90  CONTINUE
      IND(J+1,2)=M2+1
      LIST(1)=1
      J=1
      DO 110 I=1,N
        IF(IND(I,7).GT.0) J=J+1
110  LIST(I+1)=IND(J,2)
      DO 120 I=1,N
120  GRAD(I)=LIST(I+1)-LIST(I)
      DO 130 I=1,N
        F=LIST(I)
        L=LIST(I+1)-1
130  CALL SSORT2(A,2,F,L)

C -----
C   ABSCHNITT 3 (BERECHNUNG VON R(G))
C   SCHRITTE (A) UND (B), BERECHNUNG VON KAPPA1
C   IND(I,7)  SCHICHTNUMMERN VON R(G)
C   VEC(I)    ELEMENTE DER LETZTEN SCHICHT
C -----
      START=1
      VECJ=1
140  GRDMIN=N
      DO 150 I=1,N
        IF(NR(I).GT.0) GOTO 150
        IF(GRDMIN.LE.GRAD(I)) GOTO 150
        START=I
        GRDMIN=GRAD(I)
150  CONTINUE
C -----
160  G=START
      NN=N
      CALL LEVEL(G,NN,DEPTHF,K1,KAPPA1)
      J=NN-K1
      DO 180 I=1,K1
        III=I+J
180  VEC(I)=IND(III,6)
      DO 190 I=1,N
190  IND(I,7)=IND(I,8)

```

```

C -----
C   ABSCHNITT 4 (BERECHNUNG VON R(H)
C   SCHRITTE (C) BIS (F), BERECHNUNG VON KAPPA2
C   IND(I,8) SCHICHTNUMMERN VON R(H)
C -----
      LEVWTH=N
      DO 210 I=1,K1
        START=VEC(I)
        N1=N
        CALL LEVEL(START,N1,DEPTHB,K1N,KAPPA2)
        IF(DEPTHB.GT.DEPTHF) GOTO 160
        IF(KAPPA2.GE.LEVWTH) GOTO 210
        LEVWTH=KAPPA2
        VECJ=I
210  CONTINUE
      H=VEC(VECJ)
      N1=N
      CALL LEVEL(H,N1,DEPTHB,K1N,KAPPA2)
C -----
C   ABSCHNITT 5 (VORLAEUFIGE ZAEHLUNG DER ELEMENTE VON S(I))
C   SCHRITTE (G)
C   IND(I,4) VORLAEUFIGE ANZAHL DER ELEMENTE VON S(I)
C   IND(I,5) SCHICHTNUMMERN FUER KNOTEN MIT GLEICHER
C   NUMERIERUNG, SONST 0
C -----
      DO 230 I=1,N
        IND(I,4)=0
230  IND(I,5)=0
      J=0
      KNP1=DEPTHF+1
      DO 260 I=1,N
        INDI8=IND(I,8)
        IF(INDI8.EQ.0) GOTO 260
        K2=KNP1-INDI8
        IF(IND(I,7).NE.K2) GOTO 250
        IND(K2,4)=IND(K2,4)+1
        K2=-K2
        J=J+1
250  CONTINUE
        IND(I,5)=K2
260  CONTINUE
C -----
C   ABSCHNITT 6 (BESTIMMUNG UND SORTIERUNG DER V(I))
C   SCHRITTE (G)
C   VEC(I) STARTELEMENTE DER V(I) SORTIERT NACH IABS(V(I))

```

```

C -----
      K2=0
      IF(J.EQ.NN) GOTO 412
      DO 290 I=1,N
        IF(IND(I,5).LE.0) GOTO 290
        START=I
        CALL KOMPON(START,N1)
        K2=K2+1
        VEC(K2)=START
        IND(START,8)=N1
290  CONTINUE
      DO 310 I=1,N
        IF(IND(I,5).LT.-C) IND(I,5)=IND(I,5)+C2
310  CONTINUE
      CALL SSORT2(VEC,8,1,K2)
      N1M=VEC(K2)
      N1M=IND(N1M,8)
      DO 315 I=1,K2
315  IND(I,8)=VEC(I)

C -----
C      ABSCHNITT 7 (BERECHNUNG VON S)
C      SCHRITTE (H) BIS (J)
C      IND(I,4) ANZAHL DER ELEMENTE VON S(I)
C      IND(I,5) SCHICHTNUMMERN IN S
C      IND(I,6) ALLE KNOTEN V(I)
C -----
      DO 319 J=1,DEPTHF
319  VEC(J)=0
      K2P1=K2+1
      DO 410 NUE=1,K2
        III=K2P1-NUE
        START=IND(III,8)
        CALL KOMPON(START,N1)
        IND1=7
        IS=0
325  DO 330 J=1,N1
        INDJ6=IND(J,6)
        III=IND(INDJ6,IND1)+IS
        VEC(III)=VEC(III)+1
330  IND(J,2)=III
      KAPPA=0
      DO 340 J=1,N1
        INDJ2=IND(J,2)
        III=VEC(INDJ2)
        IF(III.LE.0) GOTO 340

```



```

      III=III+IND(INDJ2,4)
      VEC(INDJ2)=0
      IF(III.GT.KAPPA) KAPPA=III
340  CONTINUE
      IF(IS.GT.0) GOTO 346
      KAPPA3=KAPPA
      IND1=5
      IS=C2
      GOTO 325
C -----
346  KAPPA4=KAPPA
      IF(KAPPA3-KAPPA4) 350,347,380
347  IF(KAPPA1.GT.KAPPA2) GOTO 380
350  DO 370 J=1,N1
      INDJ6=IND(J,6)
      III=IND(INDJ6,7)
      IND(INDJ6,5)=III-C2
370  IND(III,4)=IND(III,4)+1
      GOTO 410
380  DO 400 J=1,N1
      INDJ2=IND(J,2)
400  IND(INDJ2,4)=IND(INDJ2,4)+1
410  CONTINUE
      DO 411 J=1,N1M
411  GRAD(J)=LIST(J+1)-LIST(J)
412  DO 415 I=1,N
      IF(IND(I,5).LT.-C) IND(I,5)=IND(I,5)+C2
415  IND(I,5)=IABS(IND(I,5))

C -----
C   ABSCHNITT 8 (NUMERIERUNG)
C   SCHRITTE (K) BIS (O)
C -----
      DO 420 I=1,N
420  VEC(I)=I
      CALL SSORT2(VEC,5,1,N)
      IND1=1
      L=1
      OLD=0
      NEW=1
      IND(1,7)=G
      NR(G)=NUM
      NUM=NUM+1
C -----
450  I=0
C -----

```

```

460 I=I+1
    IF(I.GT.NEW) GOTO 490
470 INDI7=IND(I,7)
    L1=LIST(INDI7)
    L2=LIST(INDI7+1)-1
    DO 480 J=L1,L2
        START=A(J)
        IF(NR(START).GT.0) GOTO 480
        IF(IND(START,5).NE.L) GOTO 480
        NR(START)=NUM
        NUM=NUM+1
        NEW=NEW+1
        IND(NEW,7)=START
480 CONTINUE
    GOTO 460
C -----
490 IF(NEW-OLD.GE.IND(L,4)) GOTO 510
    GRDMIN=N
    IND2=IND1
    DO 500 J=IND1,N
        VECJ=VEC(J)
        INDJ5=IND(VECJ,5)
        IF(INDJ5-L) 499,491,501
491 IF(NR(VECJ).GT.0) GOTO 500
        IF(GRAD(VECJ).GE.GRDMIN) GOTO 500
        GRDMIN=GRAD(VECJ)
        START=VECJ
        GOTO 500
499 IND2=J+1
500 CONTINUE
501 IND1=IND2
    NR(START)=NUM
    NUM=NUM+1
    NEW=NEW+1
    IND(NEW,7)=START
    GOTO 470
C -----
510 NEW=NEW-OLD
    DO 520 I=1,NEW
        III=I+OLD
520 IND(I,7)=IND(III,7)
    OLD=NEW
    L=L+1
    IF(L.LE.DEPTHF) GOTO 450
    IF(NUM.LE.N) GOTO 140

```

```

C -----
C   ABSCHNITT 9 (BERECHNUNG VON KNEW)
C -----
      KNEW=0
      DO 670 I=1,N
        N1=NR(I)
        L1=LIST(I)
        L2=LIST(I+1)-1
        IF(L1.GT.L2) GOTO 670
        DO 660 J=L1,L2
          K=A(J)
          III=IABS(N1-NR(K))
          IF(III.GT.KNEW) KNEW=III
660   CONTINUE
670   CONTINUE
680   IF(KOLD.GT.KNEW) GOTO 700
      KNEW=KOLD
      DO 690 I=1,N
690   A(I)=I
      RETURN
700   DO 710 I=1,N
710   A(I)=NR(I)
      RETURN
      END

C -----
C
      SUBROUTINE LEVEL(START,NN,DEPTH,K3,WIDTH)
C -----
C   ERZEUGUNG DER SCHICHTUNG R(START)
C   DEPTH   TIEFE DER SCHICHTUNG
C   K3      ANZAHL DER KNOTEN IN DER LETZTEN SCHICHT
C   WIDTH   BREITE DER SCHICHTUNG
C   NN      ANZAHL DER ZUGEORDNETEN KNOTEN
C -----
      INTEGER START,NN,DEPTH,K3,WIDTH
      INTEGER A(4096),VEC(650),
*          IND(651,8),LIST(651),GRAD(650),NR(650)
      COMMON A,VEC,IND
      EQUIVALENCE (LIST(1),IND(1,1)),(GRAD(1),IND(1,2)),
*          (NR(1),IND(1,3))
      INTEGER I,J,BEGIN,ENDE,N1,K,K2,LBR,STARTN,AI,L1,L2
      J=NN
      DO 1 I=1,J
1     IND(I,8)=0

```

```

      BEGIN=1
      N1=1
      K=1
      LBR=1
      K2=1
      IND(1,6)=START
      IND(START,8)=1
C -----
      3 K=K+1
      ENDE=N1
      DO 10 J=BEGIN,ENDE
        STARTN=IND(J,6)
        L1=IND(STARTN,1)
        L2=IND(STARTN+1,1)-1
        DO 5 I=L1,L2
          AI=A(I)
          IF(IND(AI,8).NE.0) GOTO 5
          IND(AI,8)=K
          N1=N1+1
          IND(N1,6)=AI
        5 CONTINUE
      10 CONTINUE
      K3=K2
      K2=N1-ENDE
      IF(LBR.LT.K2) LBR=K2
      BEGIN=ENDE+1
      IF(K2.GT.0) GOTO 3
C -----
      DEPTH=K-1
      WIDTH=LBR
      NN=N1
      RETURN
      END

C -----
C
      SUBROUTINE KOMPON(START,N1)
C -----
C   BERECHNUNG DER KOMPONENTE V(I), IN DER START LIEGT
C   N1           ANZAHL DER VERARBEITETEN KNOTEN
C   IND(I,6)     ALLE KNOTEN V(I)
C -----
      INTEGER START,N1
      INTEGER A(4096),VEC(650),
      *          IND(651,8),LIST(651),GRAD(650),NR(650)
      COMMON A,VEC,IND

```

```

      EQUIVALENCE (LIST(1),IND(1,1)),(GRAD(1),IND(1,2)),
*                (NR(1),IND(1,3))
      INTEGER I,J,BEGIN,ENDE,K2,STARTN,AI,L1,L2,C,C2
      DATA C/10000/
      C2=2*C
      BEGIN=1
      N1=1
      IND(START,5)=IND(START,5)-C2
      IND(1,6)=START
C -----
      3 ENDE=N1
      DO 10 J=BEGIN,ENDE
        STARTN=IND(J,6)
        L1=IND(STARTN,1)
        L2=IND(STARTN+1,1)-1
        DO 5 I=L1,L2
          AI=A(I)
          IF(IND(AI,5).LT.0) GOTO 5
          IND(AI,5)=IND(AI,5)-C2
          N1=N1+1
          IND(N1,6)=AI
        5 CONTINUE
      10 CONTINUE
      K2=N1-ENDE
      BEGIN=ENDE+1
      IF(K2.GT.0) GOTO 3
      RETURN
      END

C -----
C
      SUBROUTINE SSORT1(F,L)
C -----
C      SORTIERUNG VON A VON A(F) BIS A(L)
C -----
      INTEGER F,L
      INTEGER A(4096),VEC(650),
*      IND(651,8),LIST(651),GRAD(650),NR(650)
      COMMON A,VEC,IND
      EQUIVALENCE (LIST(1),IND(1,1)),(GRAD(1),IND(1,2)),
*                (NR(1),IND(1,3))
      INTEGER I,J,N2,S,T,IS,JS,LS,AI
      IF(L.LE.F) RETURN
      N2=(L-F+1)/2
      S=1023
      DO 100 T=1,10

```

```

        IF(S.GT.N2) GOTO 90
        LS=L-S
        DO 20 I=F,LS
            IS=I+S
            AI=A(IS)
            J=I
            JS=IS
5       IF(AI.GE.A(J)) GOTO 10
            A(JS)=A(J)
            JS=J
            J=J-S
            IF(J.GE.F) GOTO 5
10      A(JS)=AI
20      CONTINUE
90      S=S/2
100     CONTINUE
        RETURN
        END

```

```

C -----
C
C       SUBROUTINE SSORT2(VEC,K,F,L)
C -----
C       SORTIERUNG VON VEC(I), SO DASS IND(VEC(I),K) NICHT FAELLT
C       VON I=F BIS I=L
C -----
C
C       INTEGER F,L,K,VEC(650)
C       INTEGER A(4096),VECC(650),
C       *       IND(651,8),LIST(651),GRAD(650),NR(650)
C       COMMON A,VECC,IND
C       EQUIVALENCE (LIST(1),IND(1,1)),(GRAD(1),IND(1,2)),
C       *       (NR(1),IND(1,3))
C       INTEGER I,J,N2,S,T,IS,JS,LS,AI,AJ,GI
C       IF(L.LE.F) RETURN
C       N2=(L-F+1)/2
C       S=63
C       DO 100 T=1,6
C           IF(S.GT.N2) GOTO 90
C           LS=L-S
C           DO 20 I=F,LS
C               IS=I+S
C               AI=VEC(IS)
C               GI=IND(AI,K)
C               J=I
C               JS=IS
5          AJ=VEC(J)

```

```

        IF(GI.GE.IND(AJ,K)) GOTO 10
        VEC(JS)=VEC(J)
        JS=J
        J=J-S
        IF(J.GE.F) GOTO 5
10     VEC(JS)=AI
20     CONTINUE
90     S=S/2
100    CONTINUE
        RETURN
        END

```

```

C -----
C     ENDE UP
C -----

```

Zunächst erinnern wir uns an das FORTRAN-Programm TRANSFER.FOR aus [4] und [64], Kap. 3.1.2, dass in einem Vorabschritt in der Lage ist, aus der Netzstruktur von Knoten (FEM-Modell) die schwach besetzte symmetrische FEM-Matrix zu erzeugen.

```

C -----
C     HP : TRANSFER.FOR      DOS-Version
C     UP : -
C -----
C     HAUPTPROGRAMM ZUR UEBERFUEHRUNG DER ELEMENTSTRUKTUR EINES FEM-
C     NETZES IN DIE MATRIXBELEGUNG IN FORM VON NICHTNULLELEMENTEN
C     (NNE) ZWECKS WEITERER MINIMIERUNG DER BANDBREITE ODER DES
C     PROFILS
C     DAS PROGRAMM IST AUSGELEGT FUER MAXIMAL
C     NMAX  = 10000 KNOTENPUNKTE
C
C     N      = ANZAHL DER KNOTENPUNKTE, ANZAHL DER ZEILEN DER
C             MATRIX
C     NMAX   = BEGRENZUNG FUER N
C     M      = ANZAHL DER VON NULL VERSCHIEDENEN MATRIXELEMENTE
C             OBERHALB DER HAUPTDIAGONALEN
C     MMAX   = BEGRENZUNG FUER M
C
C     EINGABE
C     DATEI  MIT N, (NKNOT,NP(I),-1/) 0/
C     N
C     NKNOT  = ANZAHL DER KNOTENPUNKTE PRO ELEMENT (<=8)
C             NKNOT = 0 : SCHLUSSZEILE
C     NP(I)  = KNOTENNUMMERN PRO ELEMENT
C             NP(1)<0 : SCHLUSSZEILE FUER ELEMENTTYP
C
C     AUSGABE
C     DATEI  MIT N, M, A(I)
C     N
C     M
C     A(I), I=1(1)2*M
C             ENTHAELT DIE INDIZES DER NNE DER MATRIX OBER-
C             HALB DER HAUPTDIAGONALEN IN GEORDNETER REIHEN-
C             FOLGE I1,J1, I2,J2, ...
C -----
...

```

Den GPS-Algorithmus kann man nun sinnvoll in folgenden Gesamtablauf einbetten.

1. TRANSFER.FOR

Programm zur Überführung der Elementstruktur eines FEM-Netzes mit N Knoten in die Matrixbelegung in Form von NNE.

Eingabe: FEM-Netz als Datei

```
ELEM****.DAT:  N
                NKNOT
                NP(K),  K=1,NKNOT
                ...
                -1
                NKNOT
                NP(K),  K=1,NKNOT
                ...
                -1
                ...
                0
```

Ausgabe: NNE oberhalb der Hauptdiagonalen der symmetrischen Matrix, M Paare (i, j) angeordnet zeilenweise in aufsteigender Folge als Datei.

```
MATR****.DAT:  N  M
                A(K),  K=1,2*M
```

2. BANDRED1.FOR

Programm zur Bandbreitenreduktion einer symmetrischen Matrix der Ordnung N gemäß GPS.

Eingabe: NNE oberhalb der Hauptdiagonalen der symmetrischen Matrix, M Paare (i, j) angeordnet zeilenweise in aufsteigender Folge als Datei.

```
MATR****.DAT:  N  M
                A(K),  K=1,2*M
```

Ausgabe: Alte und neue Bandbreite sowie Umnummerierung (Permutationsvektor) als Dateien.

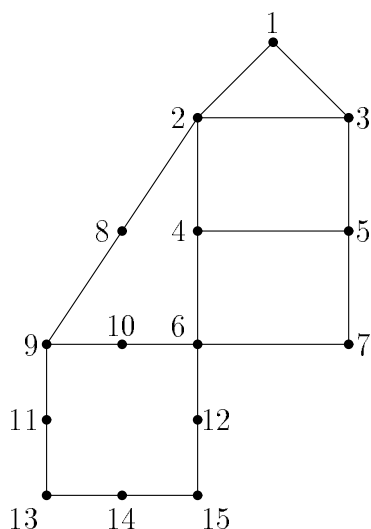
```
PERM****.DAT:  KOLD  KNEW
                N
                A(K),  K=1,N

PERMC***.DAT:  A(K),  K=1,N
```


Zur Demonstration des Gesamtablaufs kehren wir zum Beispiel [65] Bsp 1.2 zurück.

Beispiel 1.9 Gebietsvernetzung mit unterschiedlichen Elementen,
Knotenanzahl $n = 15$

Zuerst machen wir den Schritt mit dem Programm **TRANSFER.FOR**.



Netz-File **ELEM1.DAT**

```

15
3
1 2 3
-1
6
2 9 6 8 10 4
-1
4
2 4 5 3
4 6 7 5
-1
8
9 13 15 6 11 14 12 10
-1
0

```

Abb. 1.5 Gebietsvernetzung

Besetzungsstruktur der
symmetrischen Matrix $A(15, 15)$

```

x x x
x x x x x   x x x
  x x x
    x x x x x x x
      x x x
        x x x x x x x x x
          x
            x x x
              x x x x x x x
                x x x x x x
                  x x x x x
                    x x x x
                      x x x
                        x x
                          x

```

$nne = 117$ NNE mit Indizes (i, j) , $i \leq j$

```

n = 15
1  1  1  2  1  3  2  2  2  3
2  4  2  5  2  6  2  8  2  9
2 10  3  3  3  4  3  5  4  4
4  5  4  6  4  7  4  8  4  9
4 10  5  5  5  6  5  7  6  6
6  7  6  8  6  9  6 10  6 11
6 12  6 13  6 14  6 15  7  7
8  8  8  9  8 10  9  9  9 10
9 11  9 12  9 13  9 14  9 15
10 10 10 11 10 12 10 13 10 14
10 15 11 11 11 12 11 13 11 14
11 15 12 12 12 13 12 14 12 15
13 13 13 14 13 15 14 14 14 15
15 15

```

Die Bandbreiten der Ausgangsmatrix A sind $KOLD = b2 = 9$, $bw = 1 + 2 \cdot b2 = 19$.
Das Profil beträgt $15 + 71 = 86$.

Das NNE-File **MATR1.DAT** dazu mit 51 geordneten Indexpaaren (i, j) , $i < j$, ist

```

15  51
  1   2   1   3   2   3   2   4   2   5
  2   6   2   8   2   9   2  10   3   4
  3   5   4   5   4   6   4   7   4   8
  4   9   4  10   5   6   5   7   6   7
  6   8   6   9   6  10   6  11   6  12
  6  13   6  14   6  15   8   9   8  10
  9  10   9  11   9  12   9  13   9  14
  9  15  10  11  10  12  10  13  10  14
 10  15  11  12  11  13  11  14  11  15
 12  13  12  14  12  15  13  14  13  15
 14   15

```

Unter Verwendung dieses NNE-Files führt man nun den Schritt mit dem Programm **BANDRED1.FOR** aus.

Die zwei Ergebnisdateien mit dem Permutationsvektor p sind

```

perm1.dat
  9   7
 15
  1   3   2   5   4  10   7   6   9   8
 12  13  11  14  15

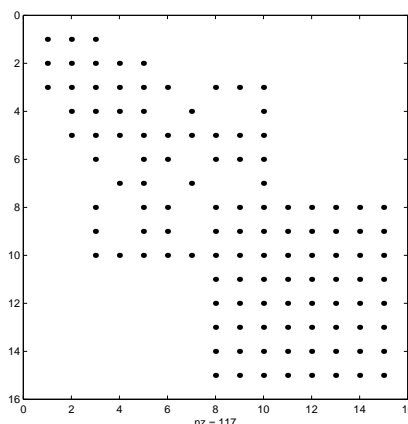
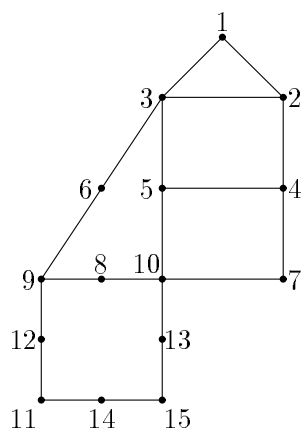
```

```

permc1.dat
  1   3   2   5   4  10   7   6   9   8
 12  13  11  14  15

```

Die Neunummerierung der Knoten der Gebietsvernetzung sowie die bandbreitenreduzierte Matrix $A' = P^T A P$ sind



$KNEW = b2 = 7$,
 $bw = 15$,
 Profil = $15 + 57 = 72$

Interessant ist ein Vergleich mit CM/RCM (Programm CUTHILL.FOR).
 Das Netz-File wurde schon notiert. Nur mit der kleinen Veränderung, dass das Programm sich als Startknoten denjenigen mit minimalen Grad selbst bestimmen soll.
 Die Ergebnisdatei RESCUTH.DAT ist

NAME DER DATEI : eelem1.dat

ALGORITHMUS VON CUTHILL-MCKEE FUER

15 KNOTENPUNKTE

KNOTENNUMMERN PRO ELEMENT

1	2	3					
2	9	6	8	10	4		
2	4	5	3				
4	6	7	5				
9	13	15	6	11	14	12	10

MINIMALER GRAD = 2 MAXIMALER GRAD = 12

MINIMALE BANDBREITE = 6

STARTNUMMERN MIT KLEINSTEM GRAD

1

ERGEBNISSE DER NEUNUMERIERUNGEN

STARTPUNKT BANDBREITE PROFIL CM PROFIL RCM

1	8	77	71
---	---	----	----

MINIMALE BANDBREITE M = 8 FUER STARTPUNKT 1

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

1	3	2	5	4	9	10	6	8	7
13	11	15	12	14					

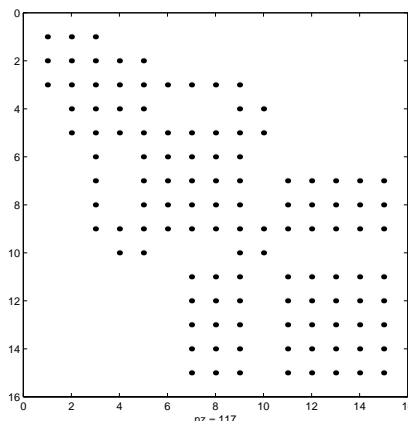
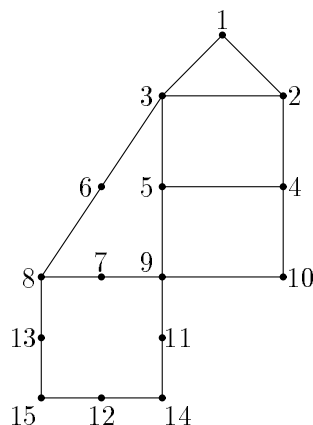
MINIMALES PROFIL = 71 FUER STARTPUNKT 1

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

15	13	14	11	12	7	6	10	8	9
3	5	1	4	2					

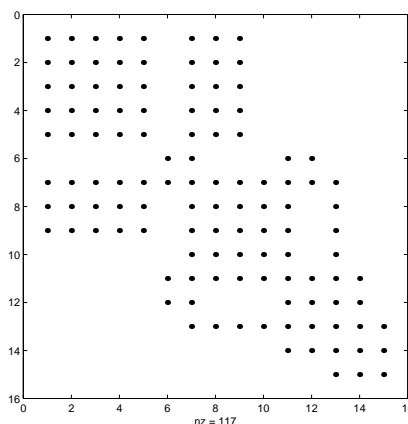
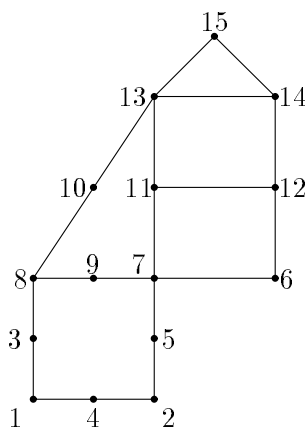
Auch hier geben wir die Neunummerierung der Knoten der Gebietsvernetzung sowie die bandbreitenreduzierte Matrix $A' = P^T A P$ an.

CM, minimale Bandbreite:



$KNEW = b_2 = 8,$
 $bw = 17,$
 $Profil = 15 + 62 = 77$

RCM, minimales Profil (Spiegelung der obigen CM-Matrix):



$KNEW = b_2 = 8,$
 $bw = 17,$
 $Profil = 15 + 56 = 71$

Natürlich kann man alle 15 Knoten als Startknoten testen, um eventuell die gleiche oder vielleicht auch eine kleinere Bandbreite wie bei GPS zu erhalten.

Das Ergebnis ist dann

NAME DER DATEI : eelem1m.dat

ALGORITHMUS VON CUTHILL-MCKEE FUER

15 KNOTENPUNKTE

MINIMALER GRAD = 2 MAXIMALER GRAD = 12

MINIMALE BANDBREITE = 6

VORGEGEBENE STARTNUMMERN

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15					

ERGEBNISSE DER NEUNUMERIERUNGEN

STARTPUNKT	BANDBREITE	PROFIL CM	PROFIL RCM
1	8	77	71
2	8	87	72
3	8	77	69
4	8	88	81
5	9	84	71
6	12	108	95
7	10	93	80
8	11	100	78
9	10	91	84
10	10	91	84
11	7	72	67
12	7	72	67
13	7	72	67
14	7	72	67
15	7	72	67

1	8	77	71
2	8	87	72
3	8	77	69
4	8	88	81
5	9	84	71
6	12	108	95
7	10	93	80
8	11	100	78
9	10	91	84
10	10	91	84
11	7	72	67
12	7	72	67
13	7	72	67
14	7	72	67
15	7	72	67

<---

MINIMALE BANDBREITE M = 7 FÜR STARTPUNKT 11

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

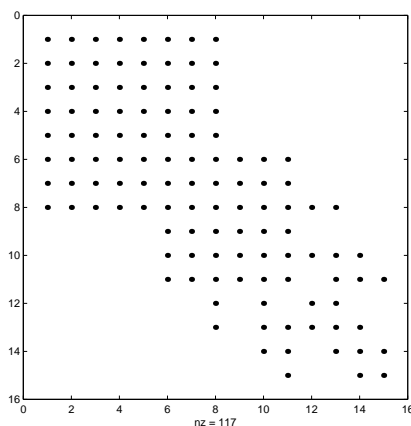
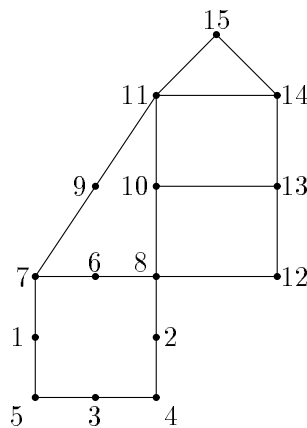
15	11	14	10	13	8	12	9	7	6
1	2	5	3	4					

MINIMALES PROFIL = 67 FÜR STARTPUNKT 11

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

1	5	2	6	3	8	4	7	9	10
15	14	11	13	12					

CM, minimale Bandbreite 7 für Startknoten 11:



$K_{NEW} = b_2 = 7,$
 $bw = 15,$
 $Prof(B_{CM}) = 72,$
 $Prof(B_{RCM}) = 67$

Bleibt noch der Vergleich mit dem MATLAB-Kommando `symrcm`.

Das Ergebnis kann hier nicht befriedigen, denn die Bandbreite wird sogar größer. Der Permutationsvektor für die Neunummerierung der Knoten gemäß

```
S=sparse(A);
p3 = symrcm(S)

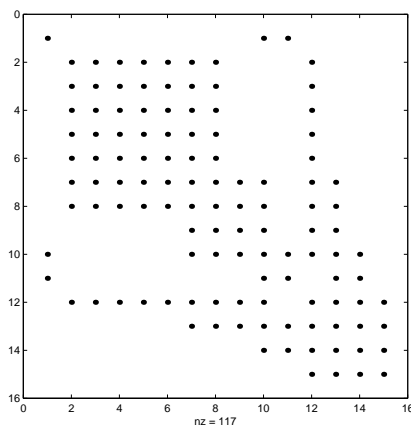
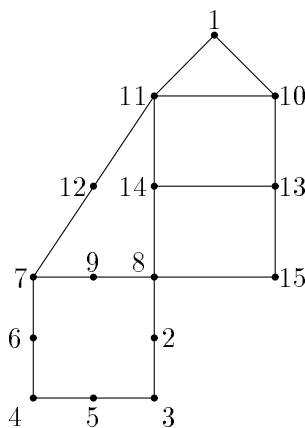
B3=S(p3,p3);
spy(B3)
print bild35.ps -dps
[i j s] = find(B3);
bw3 = 1+max(i-j)-min(i-j)    % =23
pause
```

```
% oder
P3T = zeros(n,n);
for i = 1:n
    P3T(i,p3(i)) = 1;
end;
B3 = P3T*A*P3T';

for i = 1:n
    j=p3(i);
    pp3(j)=i;
end;
pp3
```

ist $pp3 = (1, 10, 11, 13, 14, 12, 15, 9, 7, 8, 4, 6, 2, 5, 3)$.

MATLAB-Kommando `symrcm`



$b2 = 11,$
 $bw = 23,$
 $\text{Profil} = 15 + 65 = 80$

1.2 Testbeispiele zur Bandbreitenreduktion

An mehreren Beispielen zeigen wir noch die Wirkungsweise und Ergebnisse der Algorithmen von GPS (Programm BANDRED1.FOR) und CM/RCM (Programm CUT-HILL.FOR).

Beispiel 1.10 Als Modell liegt ein elliptisches Randwertproblem mit Dirichletschen Randbedingungen auf einem Rechteckgebiet zu Grunde. Die Diskretisierung ist die Finite-Differenzen-Methode auf einem äquidistanten Rechteckgitter, die mit dem 9-Punkt-Differenzenstern arbeitet.

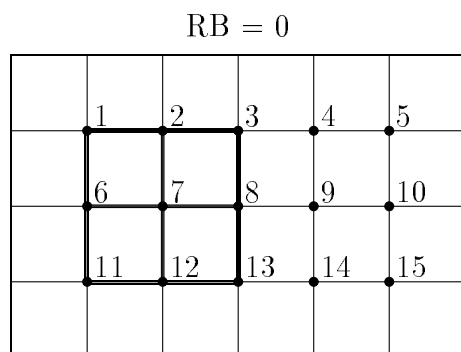


Abb. 1.6 Gitterstruktur im Lösungsgebiet (Rechteck) mit Knotennummerierung

An den inneren Punkten ist die Lösung zu bestimmen. Wenn man die Gleichungen mit dem 9-Punkt-Differenzenstern an allen diesen Punkten aufstellt, erhält man das System mit den 15 unbekannten Knotenvariablen. Gleichzeitig bilden die 15 Punkte selbst ein Vierecknetz, dass zu einer sekundären Vernetzung bei der FEM führt.

Zu diesem diskreten Modell kann man die folgenden Dateien definieren.

Netz-File **ELEM2.DAT**

```

15
4
1 2 6 7
2 3 7 8
3 4 8 9
4 5 9 10
6 7 11 12
7 8 12 13
8 9 13 14
9 10 14 15
-1
0

```

NNE-File **MATR2.DAT** mit Positionen (i, j) , $i < j$, der 38 NNE oberhalb der Hauptdiagonalen

```

15 38
1 2 1 6 1 7
2 3 2 6 2 7 2 8
3 4 3 7 3 8 3 9
4 5 4 8 4 9 4 10
5 9 5 10
6 7 6 11 6 12
7 8 7 11 7 12 7 13
8 9 8 12 8 13 8 14
9 10 9 13 9 14 9 15
10 14 10 15
11 12 12 13
13 14 14 15

```

Die Besetzungsstruktur der symmetrischen Matrix $A(n, n)$, $n = 15$, mit $nne = 91$ NNE, $nneo = 38$, ist

```

X X      X X
X X X    X X X
  X X X  X X X
    X X X  X X X
      X X    X X
X X      X X      X X
X X X    X X X    X X X
  X X X  X X X    X X X
    X X X  X X X    X X X
      X X    X X      X X
        X X      X X
          X X X    X X X
            X X X  X X X
              X X    X X

```

Auf Grund der Zeilennummerierung im Gitter hat sie eine Blocktridiagonalform mit (5×5) -Untermatrizen und der Bandbreite $KOLD = b2 = 6$. Sie entspricht der maximalen Differenz zweier benachbarter Knoten.

Dieser Adjazenzmatrix kann man auch den Graphen zuordnen.

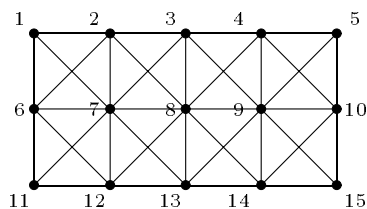


Abb. 1.7 Graph zur Gitterstruktur mit Knotennummerierung

Die Anwendung von GPS liefert die Ergebnisdatei

perm2.dat

```

    6    4
  15
    1    4    7    10   13    2    5    8    11   14
    3    6    9    12   15

```

mit dem Permutationsvektor p für die Matrixtransformation $A' = P^T A P$ zur Bandbreitenreduktion.

Damit erhält man an Stelle der zeilenweisen Nummerierung der Knoten die günstigere spaltenweise, was der schon im Beispiel 1.7 genannten Strategie entspricht.

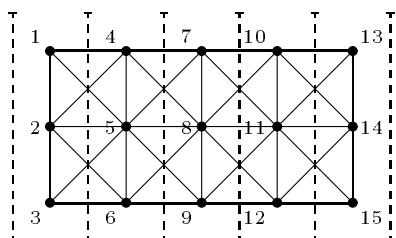


Abb. 1.8 Neunummerierung der Knoten mit GPS und Schichtung

Die Besetzungsstruktur der neuen Matrix A' ist

```

X X   X X
X X X X X X
  X X   X X
X X   X X   X X
X X X X X X X X
  X X   X X   X X
      X X   X X   X X
      X X X X X X X X
        X X   X X   X X
            X X   X X   X X
            X X X X X X X X
              X X   X X   X X
                  X X   X X
                  X X X X X X
                    X X   X X

```

Durch die Spaltennummerierung im Gitter hat man eine Blocktridiagonalform mit (3×3) -Untermatrizen und der kleineren Bandbreite $KNEW = b2 = 4$.

Führen wir noch eine Rechnung mit CM/RCM aus.

Es zeigt sich, dass hier durch die Stufenstruktur mit der 1. Stufe und ihrem einzigen Knoten als Wurzel die Bandbreitenreduktion nicht den Wert $b2 = 4$ erreichen kann. Der Test erfolgt mit allen 15 Knoten als Startknoten.

NAME DER DATEI : beisp2.dat

ALGORITHMUS VON CUTHILL-MCKEE FUER
15 KNOTENPUNKTE

KNOTENNUMMERN PRO ELEMENT

1	2	6	7
2	3	7	8
3	4	8	9
4	5	9	10
6	7	11	12
7	8	12	13
8	9	13	14
9	10	14	15

MINIMALER GRAD = 3 MAXIMALER GRAD = 8
MINIMALE BANDBREITE = 4

VORGEGEBENE STARTNUMMERN

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15					

ERGEBNISSE DER NEUNUMERIERUNGEN

STARTPUNKT BANDBREITE PROFIL CM PROFIL RCM

1	5	64	60	<---
2	7	83	71	
3	9	96	75	
4	6	76	66	
5	5	64	60	
6	5	63	62	
7	8	81	68	
8	9	102	86	
9	8	77	65	
10	5	63	62	
11	6	70	62	
12	7	83	71	
13	9	96	75	
14	6	76	66	
15	6	70	62	

MINIMALE BANDBREITE $M = 5$ FUER STARTPUNKT 1
 DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

1	3	7	10	13	2	4	8	11	14
5	6	9	12	15					

MINIMALES PROFIL = 60 FUER STARTPUNKT 1
 DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

15	13	9	6	3	14	12	8	5	2
11	10	7	4	1					

Nehmen wir also den Permutationsvektor für den Startpunkt 1, um die Unnummerierung der Knoten durchzuführen.

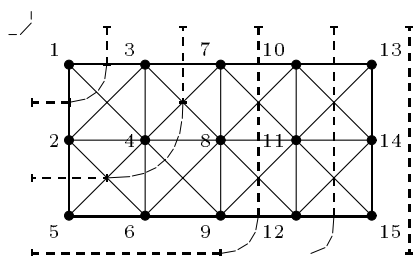


Abb. 1.9 Neunummerierung der Knoten mit CM/RCM und Schichtung

Der maximale Abstand benachbarter Knoten im Graphen beträgt 5, gebildet durch die Knotenpaare (3,8) und (4,9). Diese sind zuständig für die Bandbreite 5.

Die Besetzungsstruktur der neuen Matrix A' ist

X	X	X	X																
X	X	X	X	X	X														
X	X	X	X			X	X												
X	X	X	X	X	X	X	X	X	X										
	X		X	X	X														
	X		X	X	X		X	X											
		X	X			X	X		X	X									
		X	X		X	X	X	X	X	X	X	X							
			X		X		X	X		X	X								
				X	X		X	X		X	X		X	X					
					X	X	X	X	X	X	X	X							
						X	X		X	X		X	X						
							X	X		X	X		X	X					
								X	X	X	X	X	X						
									X	X		X	X						

Durch die Stufenstruktur unterscheidet sich diese Matrix von der durch GPS erzeugten im linken oberen Bereich.

Beispiel 1.11 Wir betrachten die sparse Matrix mit der Besetzungsstruktur

$$A(5,5) = \begin{pmatrix} * & & & & * \\ & * & & & \\ & & * & & \\ & & & * & \\ * & & & & * \end{pmatrix}.$$

Es gilt $bw(A) = b1(A) = 9$, $KOLD = b2(A) = 4$, $nne = 7$.

Durch genaues Hinschauen erkennt man, dass durch Zeilen- und Spaltenvertauschung $2 \leftrightarrow 5$ die Bandbreite reduziert wird.

*			*		*	*				*	*
	*				*	*	*			*	*
		*				*					*
			*				*			*	
*				*		*					*

Der Permutationsvektor für die zugehörige Transformation $A' = P^T A P$ ist $\tilde{p} = (1, 5, 3, 4, 2)$. Damit ist $bw(A') = b1(A') = 3$, $KNEW = b2(A') = 1$.

Untersuchen wir das Verhalten von GPS.

Ein Netz-File zu dieser Matrix existiert nicht.

Das NNE-File **MATR3.DAT** mit Positionen (i, j) , $i < j$, des einzigen NNE oberhalb der Hauptdiagonalen ist

```
5 1
1 5
```

Die Anwendung von GPS ergibt die Ergebnisdatei

```
perm3.dat
4      1
5
4      1      2      3      5
```

Der Permutationsvektor $p = (4, 1, 2, 3, 5)$ hier unterscheidet sich von obigen Vektor \tilde{p} . Es werden mehr Vertauschungen durchgeführt, die wir schrittweise nachvollziehen wollen.

```

          1  2  3  4  5
Tausch 1: 1 ↔ 2    2  1  3  4  5
Tausch 2: 3 ↔ 4    2  1  4  3  5
Tausch 3: 2 ↔ 4    4  1  2  3  5 = p
```

Für die 3 Schritte der Zeilen- und Spaltenvertauschung erhält man

A	Zeilent. $P^T A$	Spaltent. AP	
$\begin{matrix} * & & & * \\ & * & & \\ & & * & \\ & & & * \\ * & & & * \end{matrix}$	$\begin{matrix} * & & & \\ * & & & * \\ & * & & \\ & & * & \\ * & & & * \end{matrix}$	$\begin{matrix} * & & & * \\ & * & & \\ & & * & \\ & & & * \\ * & & & * \end{matrix}$	Tausch $1 \leftrightarrow 2$ $2\ 1\ 3\ 4\ 5$ $P_1^T A P_1$
	$\begin{matrix} * & & & \\ & * & & * \\ & & * & \\ & & & * \\ * & & & * \end{matrix}$	$\begin{matrix} * & & & * \\ & * & & \\ & & * & \\ & & & * \\ * & & & * \end{matrix}$	Tausch $3 \leftrightarrow 4$ $2\ 1\ 4\ 3\ 5$ $P_2^T P_1^T A P_1 P_2$
	$\begin{matrix} * & & & \\ & & * & \\ & & & * \\ & * & & * \\ * & & & * \end{matrix}$	$\begin{matrix} * & & & \\ & * & & \\ & & * & \\ & & & * \\ * & & & * \end{matrix}$	Tausch $2 \leftrightarrow 4$ $4\ 1\ 2\ 3\ 5$ $P_3^T P_2^T P_1^T A P_1 P_2 P_3$

A'

Somit ist

$$A' = P^T A P, \text{ wobei } P = \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \\ & & 1 \\ & & & 1 \end{pmatrix} \neq P^T.$$

Wie man erkennt, ist der 2. Schritt (Tausch $3 \leftrightarrow 4$) überflüssig, so dass auch der Permutationsvektor $(4, 1, 3, 2, 5)$ verwendet werden kann.

Beispiel 1.12 Gegeben sei die sparse Matrix mit der Besetzungsstruktur

$$A(6,6) = \begin{pmatrix} * & * & & & & * \\ * & * & * & & & \\ & * & * & * & & \\ & & * & * & * & \\ & & & * & * & * \\ * & & & & * & * \end{pmatrix}.$$

Es gilt $bw(A) = b1(A) = 11$, $KOLD = b2(A) = 5$, $nne = 18$.

Durch einige Überlegung erkennt man, dass durch systematische Zeilen- und Spaltenvertauschung die Bandbreite stark reduziert wird.

$$A' = P^T A P = \begin{pmatrix} * & * & * & & & \\ * & * & & & * & \\ * & & * & & * & \\ & * & & * & & * \\ & & * & & * & * \\ & & & * & * & * \end{pmatrix},$$

$$\text{wobei } P = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ 1 & & & & & 1 \end{pmatrix} \neq P^T.$$

Der Permutationsvektor für die zugehörige Transformation ist $p = (1, 3, 5, 6, 4, 2)$.

P^T wird somit durch den Permutationsvektor $\tilde{p} = (1, 6, 2, 5, 3, 4)$ erzeugt.

Damit ist $bw(A') = b1(A') = 5$, $KNEW = b2(A') = 2$.

Untersuchen wir das Verhalten von GPS.

Ein Netz-File zu dieser Matrix existiert nicht, es sei denn, einige Elemente a_{ij} sind noch Null.

Das NNE-File **MATR4.DAT** mit Positionen (i, j) , $i < j$, der 6 NNE oberhalb der Hauptdiagonalen ist

```
6 6
1 2 1 6
2 3
3 4
4 5
5 6
```

Die Anwendung von GPS mit dem Programm BANDRED1.FOR ergibt die Ergebnisdatei

perm4.dat

```

      5      2
      6
      1      2      4      6      5      3

```

Dieser Permutationsvektor ist zwar anders als der obige, aber es entsteht damit dieselbe Matrix A' .

Die Bandbreitenreduktion für die gegebene Matrixstruktur kann auf eine beliebige Dimension n verallgemeinert werden.

Es gilt für $A(n, n)$

$$p = \begin{cases} (1, 3, \dots, n-3, n-1, n, n-2, \dots, 4, 2), & \text{falls } n \text{ gerade,} \\ (1, 3, \dots, n-2, n, n-1, n-3, \dots, 4, 2), & \text{falls } n \text{ ungerade.} \end{cases}$$

Eventuell ist die Notation des Permutationsvektors für P^T günstiger, denn seine aufeinanderfolgenden Komponenten bilden dann Paare der Gestalt $(k, n+1-k)$.

$$\tilde{p} = \begin{cases} (1, n, 2, n-1, \dots, n/2, n/2+1), & \text{falls } n \text{ gerade,} \\ (1, n, 2, n-1, \dots, (n-1)/2, (n+1)/2+1, (n+1)/2), & \text{falls } n \text{ ungerade.} \end{cases}$$

Steht in der Matrix A das Symbol $*$ für einen Block, dann kann man in der Permutationsmatrix P die Eins als Einheitsblock nehmen.

Beispiel 1.13 Dreiecksvernetzung eines dreieckigen Gebietes mit Loch,
 $n = 6$, $nne = 30$.

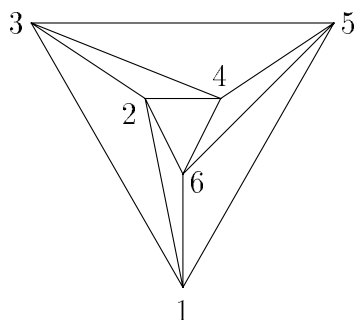


Abb. 1.10 Dreiecksvernetzung

Besetzungsstruktur der
symmetrischen Matrix $A(6,6)$

x	x	x		x	x
		x	x	x	
			x	x	x
				x	x
					x

Netz-File **ELEM5.DAT**

```
6
3
1 2 3
1 2 6
2 3 4
3 4 5
5 4 6
5 6 1
-1
0
```

NNE-File **NMATR5.DAT**

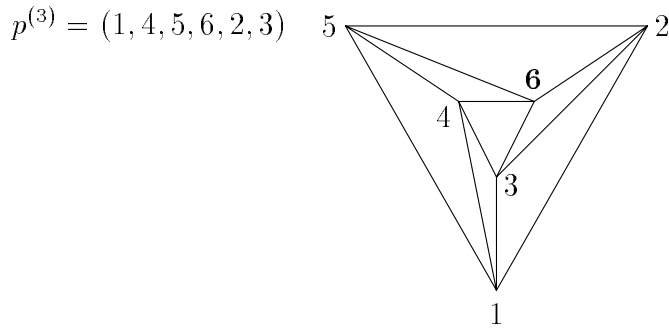
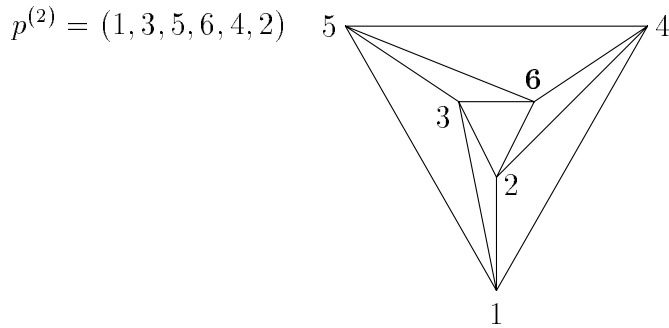
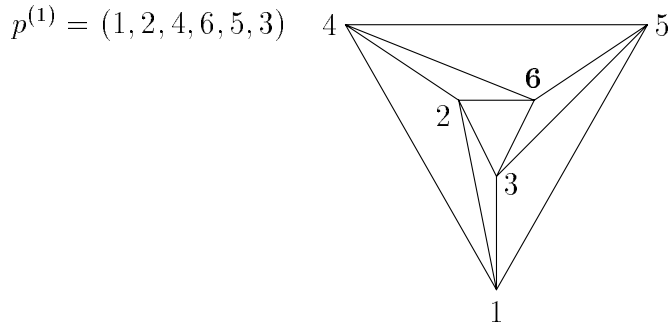
```
1
6 30 12
1 1 1 2 1 3 1 5 1 6
2 2 2 3 2 4 2 6 3 3
3 4 3 5 4 4 4 5 4 6
5 5 5 6 6 6
```

Das NNE-File **MATR5.DAT** mit Positionen (i, j) , $i < j$, der 12 NNE oberhalb der Hauptdiagonalen ist

```
6 12
1 2 1 3 1 5 1 6
2 3 2 4 2 6
3 4 3 5
4 5 4 6
5 6
```

Da jeder Knoten mit den anderen bis auf einen verbunden ist, hat jeder den Grad 4. Man erkennt eine mögliche Strategie der Umnummerierung. So sollte dann der Knoten 1 nur die Knoten 2,3,4,5 als Nachbarn haben. Somit ist entscheidend, dass in Abb. 1.10 der mit dem Knoten 1 nicht verknüpfte Knoten 4 auf jeden Fall die neue Nummer 6 erhält. Die Knoten 1 und 6 dürfen nicht zu einem Dreieck gehören. Hinsichtlich der anderen Nummern sind Unterschiede möglich.

Wir testen zunächst drei Permutationsvektoren.



In zweiten Fall ist die Konstruktion der Komponenten von $p^{(2)}$ wie im Beispiel 1.12. Die bandbreitenreduzierte Matrix $A' = P^T A P$ hat in den beiden ersten Fällen sowie im Fall 3 die Gestalten

$$A'_{1,2} = \begin{pmatrix} x & x & x & x & x & \\ x & x & x & x & & x \\ x & x & x & & x & x \\ x & x & & x & x & x \\ x & & x & x & x & x \\ & x & x & x & x & x \end{pmatrix}, \quad A'_3 = \begin{pmatrix} x & x & x & x & x & \\ x & x & x & & x & x \\ x & x & x & x & & x \\ x & & x & x & x & x \\ x & x & & x & x & x \\ & x & x & x & x & x \end{pmatrix},$$

und damit die um Eins kleinere Bandbreite $\text{KNEW} = b_2(A') = 4$.

Die Anwendung von GPS (BANDRED1.FOR) ergibt die Ergebnisdatei

perm5.dat

```

      5      4
      6
      1      2      3      6      4      5

```

Der Permutationsvektor ist $p = (1, 2, 3, 6, 4, 5)$ und A' hat die Form A'_3 .

Im Vergleich dazu geben wir noch CM (CUTHILL.FOR) mit seinen Dateien an, der auf $p = (1, 4, 3, 6, 2, 5)$ und ebenfalls A'_3 führt.

Das Profil von A' ist in allen Fällen $6 + 14 = 20$.

eelem5.dat

```

      6 -1
      3
      1 6 2
      1 2 3
      2 4 3
      3 4 5
      4 6 5
      6 1 5
     -1/
      0/

```

rescuth.dat

NAME DER DATEI : eelem5.dat

ALGORITHMUS VON CUTHILL-MCKEE FUER

6 KNOTENPUNKTE

KNOTENNUMMERN PRO ELEMENT

```

      1      6      2
      1      2      3
      2      4      3
      3      4      5
      4      6      5
      6      1      5

```

MINIMALER GRAD = 4 MAXIMALER GRAD = 4

MINIMALE BANDBREITE = 2

STARTNUMMERN MIT KLEINSTEM GRAD

1

ERGEBNISSE DER NEUNUMERIERUNGEN

STARTPUNKT BANDBREITE PROFIL CM PROFIL RCM

1 4 20 20

MINIMALE BANDBREITE M = 4 FUER STARTPUNKT 1

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

1 4 3 6 2 5

MINIMALES PROFIL = 20 FUER STARTPUNKT 1

DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

6 3 4 1 5 2

Die Positionen der NE in der Matrix verschieben sich durch die Neunummerierung im Allgemeinen etwas “nach außen”.

In A waren oberhalb der Diagonalen die Elemente a_{14}, a_{25}, a_{36} nicht belegt. Die Summe ihrer Indizes ist $S_{NE(A)} = \sum_{a_{ij}=0, i < j} i + j = 21$.

Nach Neunummerierung erhält man $S_{NE(A')} = \sum_{a_{ij}=0, i < j} i + j = 21 \geq S_{NE(A)}$.

Auf Grund der Netzstruktur und des generellen Knotengrades 4 ist jedoch die Matrixform

$$A' = \begin{pmatrix} x & x & x & x & & \\ x & x & x & x & x & \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ & x & x & x & x & x \\ & & x & x & x & x \end{pmatrix}$$

mit der Bandbreite $b2(A') = 3$ nicht machbar.

Beispiel 1.14 Wir betrachten das Grundgebiet des Autolängsschnittes und seine Vernetzung.

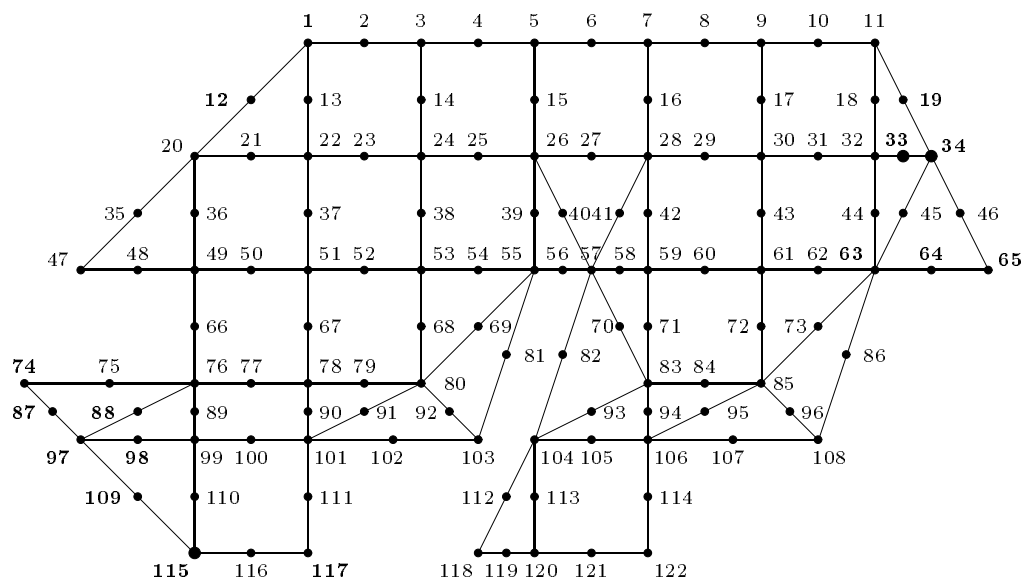


Abb. 1.11 Längsschnitt eines Autoinnenraums,
Dreieck- und Rechteckelemente mit quadratischem Ansatz,
Knotenpunkte zeilenweise durchnummeriert, $n = 122$, $nne = 1390$

Nun wenden wir die einzelnen Programme an.

(1) TRANSFER.FOR

Netz-File ELEM61.DAT → NNE-File MATR61.DAT der symmetrischen Matrix

ELEM61.DAT

```

122
6
47 49 20 48 36 35
74 97 76 87 88 75
97 99 76 98 89 88
97 115 99 109 110 98
20 22 1 21 13 12
101 80 78 91 79 90
101 103 80 102 92 91
80 55 53 69 54 68
80 103 55 92 81 69
55 57 26 56 40 39
26 57 28 40 41 27
57 59 28 58 42 41
57 83 59 70 71 58

```

```

104 83 57 93 70 82
104 106 83 105 94 93
118 120 104 119 113 112
106 85 83 95 84 94
106 108 85 107 96 95
85 63 61 73 62 72
85 108 63 96 86 73
32 34 11 33 19 18
32 63 34 44 45 33
63 65 34 64 46 45
-1/
8
49 51 22 20 50 37 21 36
76 78 51 49 77 67 50 66
99 101 78 76 100 90 77 89
115 117 101 99 116 111 100 110
22 24 3 1 23 14 2 13
51 53 24 22 52 38 23 37
78 80 53 51 79 68 52 67
24 26 5 3 25 15 4 14
53 55 26 24 54 39 25 38
26 28 7 5 27 16 6 15
120 122 106 104 121 114 105 113
28 30 9 7 29 17 8 16
59 61 30 28 60 43 29 42
83 85 61 59 84 72 60 71
30 32 11 9 31 18 10 17
61 63 32 30 62 44 31 43
-1/
0/

```

Oberhalb der Hauptdiagonalen der symmetrischen Matrix A stehen $nneo = 634$ NNE. Damit ist $nne = n + 2 \cdot nneo = 1390$.

MATR61.DAT

```

122 634
1 2 1 3 1 12 1 13 1 14 1 20 1 21 1 22 1 23 1 24
2 3 2 13 2 14 2 22 2 23 2 24 3 4 3 5 3 13 3 14
3 15 3 22 3 23 3 24 3 25 3 26 4 5 4 14 4 15 4 24
4 25 4 26 5 6 5 7 5 14 5 15 5 16 5 24 5 25 5 26
5 27 5 28 6 7 6 15 6 16 6 26 6 27 6 28 7 8 7 9
7 15 7 16 7 17 7 26 7 27 7 28 7 29 7 30 8 9 8 16
8 17 8 28 8 29 8 30 9 10 9 11 9 16 9 17 9 18 9 28
9 29 9 30 9 31 9 32 10 11 10 17 10 18 10 30 10 31 10 32
11 17 11 18 11 19 11 30 11 31 11 32 11 33 11 34 12 13 12 20
12 21 12 22 13 14 13 20 13 21 13 22 13 23 13 24 14 15 14 22
14 23 14 24 14 25 14 26 15 16 15 24 15 25 15 26 15 27 15 28
16 17 16 26 16 27 16 28 16 29 16 30 17 18 17 28 17 29 17 30
17 31 17 32 18 19 18 30 18 31 18 32 18 33 18 34 19 32 19 33
19 34 20 21 20 22 20 35 20 36 20 37 20 47 20 48 20 49 20 50

```

20	51	21	22	21	36	21	37	21	49	21	50	21	51	22	23	22	24	22	36
22	37	22	38	22	49	22	50	22	51	22	52	22	53	23	24	23	37	23	38
23	51	23	52	23	53	24	25	24	26	24	37	24	38	24	39	24	51	24	52
24	53	24	54	24	55	25	26	25	38	25	39	25	53	25	54	25	55	26	27
26	28	26	38	26	39	26	40	26	41	26	53	26	54	26	55	26	56	26	57
27	28	27	40	27	41	27	57	28	29	28	30	28	40	28	41	28	42	28	43
28	57	28	58	28	59	28	60	28	61	29	30	29	42	29	43	29	59	29	60
29	61	30	31	30	32	30	42	30	43	30	44	30	59	30	60	30	61	30	62
30	63	31	32	31	43	31	44	31	61	31	62	31	63	32	33	32	34	32	43
32	44	32	45	32	61	32	62	32	63	33	34	33	44	33	45	33	63	34	44
34	45	34	46	34	63	34	64	34	65	35	36	35	47	35	48	35	49	36	37
36	47	36	48	36	49	36	50	36	51	37	38	37	49	37	50	37	51	37	52
37	53	38	39	38	51	38	52	38	53	38	54	38	55	39	40	39	53	39	54
39	55	39	56	39	57	40	41	40	55	40	56	40	57	41	42	41	57	41	58
41	59	42	43	42	57	42	58	42	59	42	60	42	61	43	44	43	59	43	60
43	61	43	62	43	63	44	45	44	61	44	62	44	63	45	46	45	63	45	64
45	65	46	63	46	64	46	65	47	48	47	49	48	49	49	50	49	51	49	66
49	67	49	76	49	77	49	78	50	51	50	66	50	67	50	76	50	77	50	78
51	52	51	53	51	66	51	67	51	68	51	76	51	77	51	78	51	79	51	80
52	53	52	67	52	68	52	78	52	79	52	80	53	54	53	55	53	67	53	68
53	69	53	78	53	79	53	80	54	55	54	68	54	69	54	80	55	56	55	57
55	68	55	69	55	80	55	81	55	92	55	103	56	57	57	58	57	59	57	70
57	71	57	82	57	83	57	93	57	104	58	59	58	70	58	71	58	83	59	60
59	61	59	70	59	71	59	72	59	83	59	84	59	85	60	61	60	71	60	72
60	83	60	84	60	85	61	62	61	63	61	71	61	72	61	73	61	83	61	84
61	85	62	63	62	72	62	73	62	85	63	64	63	65	63	72	63	73	63	85
63	86	63	96	63	108	64	65	66	67	66	76	66	77	66	78	67	68	67	76
67	77	67	78	67	79	67	80	68	69	68	78	68	79	68	80	69	80	69	81
69	92	69	103	70	71	70	82	70	83	70	93	70	104	71	72	71	83	71	84
71	85	72	73	72	83	72	84	72	85	73	85	73	86	73	96	73	108	74	75
74	76	74	87	74	88	74	97	75	76	75	87	75	88	75	97	76	77	76	78
76	87	76	88	76	89	76	90	76	97	76	98	76	99	76	100	76	101	77	78
77	89	77	90	77	99	77	100	77	101	78	79	78	80	78	89	78	90	78	91
78	99	78	100	78	101	79	80	79	90	79	91	79	101	80	81	80	90	80	91
80	92	80	101	80	102	80	103	81	92	81	103	82	83	82	93	82	104	83	84
83	85	83	93	83	94	83	95	83	104	83	105	83	106	84	85	84	94	84	95
84	106	85	86	85	94	85	95	85	96	85	106	85	107	85	108	86	96	86	108
87	88	87	97	88	89	88	97	88	98	88	99	89	90	89	97	89	98	89	99
89	100	89	101	90	91	90	99	90	100	90	101	91	92	91	101	91	102	91	103
92	101	92	102	92	103	93	94	93	104	93	105	93	106	94	95	94	104	94	105
94	106	95	96	95	106	95	107	95	108	96	106	96	107	96	108	97	98	97	99
97	109	97	110	97	115	98	99	98	109	98	110	98	115	99	100	99	101	99	109
99	110	99	111	99	115	99	116	99	117	100	101	100	110	100	111	100	115	100	116
100	117	101	102	101	103	101	110	101	111	101	115	101	116	101	117	102	103	104	105
104	106	104	112	104	113	104	114	104	118	104	119	104	120	104	121	104	122	105	106
105	113	105	114	105	120	105	121	105	122	106	107	106	108	106	113	106	114	106	120
106	121	106	122	107	108	109	110	109	115	110	111	110	115	110	116	110	117	111	115
111	116	111	117	112	113	112	118	112	119	112	120	113	114	113	118	113	119	113	120
113	121	113	122	114	120	114	121	114	122	115	116	115	117	116	117	118	119	118	120
119	120	120	121	120	122	121	122												

(2) GPS, BANDRED1.FOR

Bandbreitenreduktion der Matrix A gegeben durch MATR61.DAT \rightarrow Permutationsvektor PERM61.DAT zur Umnummerierung

PERM61.DAT

48 25

122									
76	75	62	61	44	42	29	27	12	10
3	92	77	63	45	28	11	4	1	94
93	79	78	65	64	46	43	31	30	14
13	6	2	5	109	95	80	69	67	47
48	32	19	8	7	16	110	108	97	96
82	81	71	68	70	66	50	49	34	33
20	18	9	15	17	98	90	84	83	51
36	25	22	113	111	100	99	91	89	85
86	52	37	35	26	21	112	114	105	103
102	87	53	41	39	24	116	115	107	106
104	101	88	55	54	40	38	23	122	121
117	72	60	58	120	118	119	73	74	59
56	57								

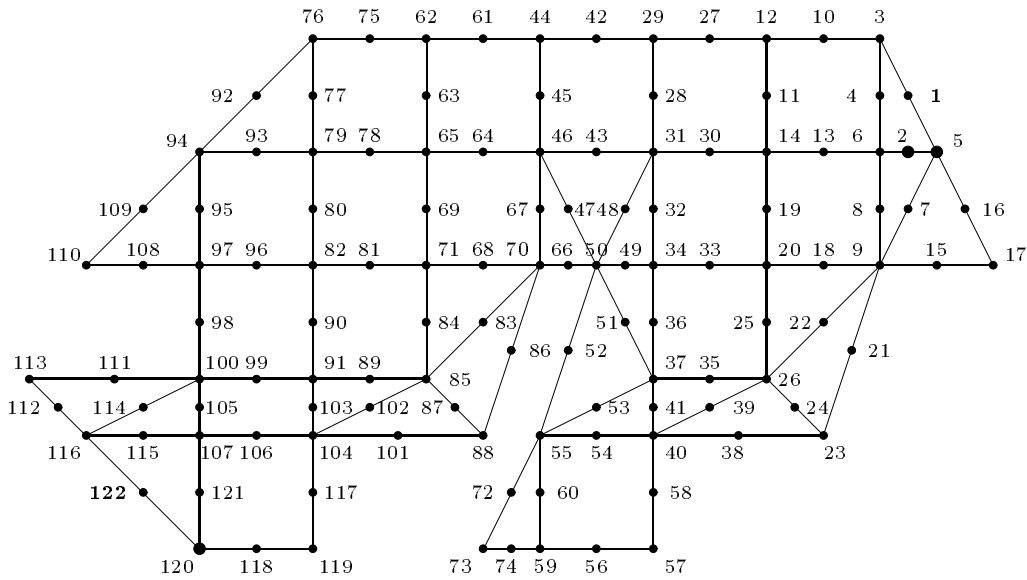


Abb. 1.12 Längsschnitt eines Autoinnenraums,
Knotenpunkte nach Umnummerierung, $n = 122$, $nne = 1390$

Die neue Bandbreite $KNEW = b_2(A') = 25$ ist nicht die erreichbare kleinste Bandbreite.

(3) CM/RCM, CUTHILL.FOR

Rechnungen dazu liegen im Beispiel [65] Bsp. 1.9 vor.

Mit der Eingabedatei BEISP61.DAT wurden mehrere Knoten als Startknoten getestet und folgende Ergebnisse erzielt.

1. Die minimale Bandbreite $b_2(A') = 21$ wird für den Startknoten 34 erreicht.

MINIMALE BANDBREITE M = 21 FUER STARTPUNKT 34
 DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

83	81	67	64	50	46	37	35	20	18
10	96	82	65	49	36	19	9	5	98
97	85	84	68	66	51	48	31	28	17
15	11	7	1	113	99	86	70	62	47
42	27	14	8	6	2	115	114	101	100
88	87	71	69	63	61	44	43	30	29
16	13	12	3	4	102	90	79	78	45
32	25	24	118	116	104	103	91	89	80
75	52	34	33	26	21	117	120	105	95
93	76	53	41	39	22	121	119	107	106
94	92	77	55	54	40	38	23	122	111
108	72	59	56	112	109	110	74	73	60
57	58								

Der Startknoten 65 liefert die Bandbreite $b_2 = 23$.

2. Das minimale Profil $Prof = 1263$ erhält man mit dem Startknoten 115, d. h. CM mit $115 \rightarrow 1$ und durch RCM wird $1 \rightarrow 122$.

MINIMALES PROFIL = 1263 FUER STARTPUNKT 115
 DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

65	64	62	61	59	58	54	43	40	20
18	67	66	63	60	55	42	19	7	81
83	80	76	74	72	71	56	53	41	39
17	16	6	5	79	84	82	75	73	69
57	47	38	15	4	3	77	78	93	95
92	90	89	85	86	70	68	49	45	37
36	14	13	2	1	96	94	91	87	50
48	32	12	106	108	109	104	103	98	97
88	52	44	31	30	8	107	111	110	105
99	101	51	35	21	10	115	117	113	114
112	102	100	46	34	33	11	9	121	116
120	29	23	26	122	119	118	27	28	22
25	24								

3. Der vorgegebene Starknoten 19 führt auf die Bandbreite $b_2 = 25$.

MINIMALE BANDBREITE M = 25 FUER STARTPUNKT 19
 DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

77	75	64	61	45	42	30	27	13	10
4	92	76	62	44	28	11	3	1	94
93	79	78	65	65	46	43	31	29	14
12	6	2	5	108	95	80	69	68	49
48	32	19	8	7	15	110	109	97	96
82	81	71	67	70	66	50	47	34	33
20	18	9	16	17	98	90	84	83	51
35	25	24	113	111	100	99	91	89	85
86	52	37	35	26	21	112	115	105	103
102	87	53	41	39	22	116	114	107	106
104	101	88	55	54	40	38	23	122	120
117	72	59	56	121	118	119	74	73	60
57	58								

Diese Situation ist vergleichbar mit dem Ablauf TRANSFER.FOR & BANDRED1.FOR. Zwischen beiden Permutationsvektoren sind nur wenige Vertauschungen bei Nachbarknoten erkennbar.

4. Der erste Startpunkt mit dem kleinsten Grad in der Knotenpunktfolge 1-122 ist der Knoten 12 mit dem Grad 5. Dieser führt jedoch auf die große Bandbreite $b_2 = 33$.
5. Die Knoten 19,33,34,63,64,65 liegen in Nachbarschaft, daher die durchaus vergleichbaren Bandbreiten.

ERGEBNISSE DER NEUNUMERIERUNGEN

STARTPUNKT	BANDBREITE
------------	------------

19	25
33	21
34	21
63	26
64	23
65	23

Beispiel 1.15 Anwendung einer FEM-Diskretisierung von Beispiel [65] Bsp. 1.5

Gegeben sei das ringförmige Gebiet mit der symmetrischen Triangulierung, also eine Dreiecksvernetzung, und Nummerierung der Knotenpunkte.

Hier sind die Knoten radial im Gegenuhrzeigersinn nummeriert.

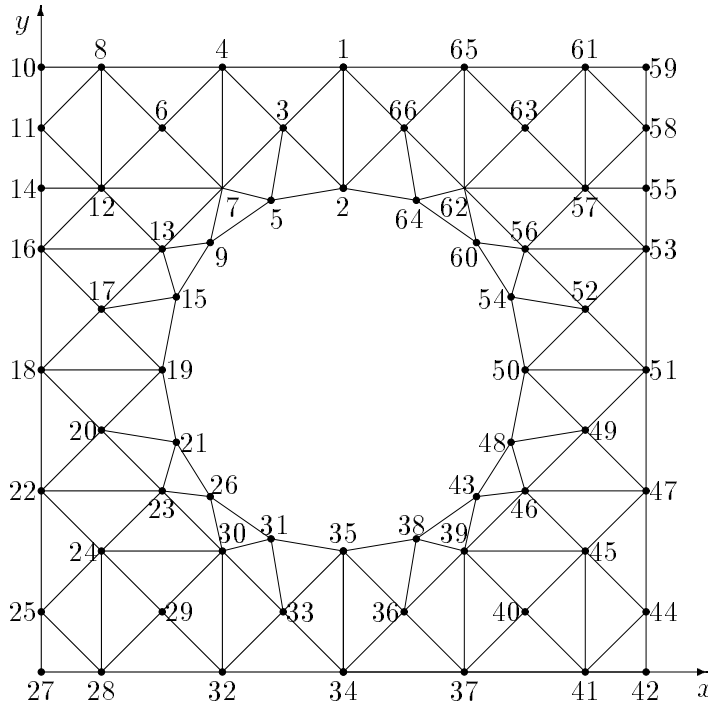


Abb. 1.13 Gebietsvernetzung mit 90 Dreieckselementen bei 66 Knoten

In der zugehörigen Adjazenzmatrix A sind $nneo = 156$ NNE oberhalb der Hauptdiagonalen und insgesamt $nne = n + 2 \cdot nneo = 378$ NNE. Ihre Bandbreite ist $b2(A) = 65$, weil die Knotenpunkte 1 und 2 mit den Knoten 64 bis 66 durch Dreieckseiten verbunden sind.

Zur Anwendung kommen drei Programme.

(1) TRANSFER.FOR

Netz-File ELEM61N.DAT \rightarrow NNE-File MATR61N.DAT der symmetrischen Matrix

ELEM61N.DAT

```

66
3
1 3 2      1 4 3      2 3 5      3 4 7      3 7 5      4 6 7      4 8 6
5 7 9      6 8 12     6 12 7     7 12 13    7 13 9     8 10 11    8 11 12
9 13 15    11 14 12    12 14 16    12 16 13    13 16 17    13 17 15    15 17 19
16 18 17   17 18 19   18 22 20   18 20 19   19 20 21   20 22 23   20 23 21
21 23 26   22 24 23   22 25 24   23 24 30   23 30 26   24 25 28   24 28 29

```

```

24 29 30 25 27 28 26 30 31 28 32 29 29 32 30 30 32 33 30 33 31
31 33 35 32 34 33 33 34 35 34 37 36 34 36 35 35 36 38 36 37 39
36 39 38 37 41 40 37 40 39 38 39 43 39 40 45 39 45 46 39 46 43
40 41 45 41 42 44 41 44 45 43 46 48 44 47 45 45 47 46 46 47 49
46 49 48 47 51 49 48 49 50 49 51 50 50 51 52 50 52 54 51 53 52
52 53 56 52 56 54 53 55 57 53 57 56 54 56 60 55 58 57 56 57 62
56 62 60 57 58 61 57 61 63 57 63 62 58 59 61 60 62 64 61 65 63
62 63 65 62 65 66 62 66 64 64 66 2 65 1 66 66 1 2
-1/
0/

```

MATR61N.DAT

```

66 156
 1  2  1  3  1  4  1 65  1 66
 2  3  2  5  2 64  2 66  3  4
 3  5  3  7  4  6  4  7  4  8
 5  7  5  9  6  7  6  8  6 12
 7  9  7 12  7 13  8 10  8 11
 8 12  9 13  9 15 10 11 11 12
11 14 12 13 12 14 12 16 13 15
13 16 13 17 14 16 15 17 15 19
16 17 16 18 17 18 17 19 18 19
18 20 18 22 19 20 19 21 20 21
20 22 20 23 21 23 21 26 22 23
22 24 22 25 23 24 23 26 23 30
24 25 24 28 24 29 24 30 25 27
25 28 26 30 26 31 27 28 28 29
28 32 29 30 29 32 30 31 30 32
30 33 31 33 31 35 32 33 32 34
33 34 33 35 34 35 34 36 34 37
35 36 35 38 36 37 36 38 36 39
37 39 37 40 37 41 38 39 38 43
39 40 39 43 39 45 39 46 40 41
40 45 41 42 41 44 41 45 42 44
43 46 43 48 44 45 44 47 45 46
45 47 46 47 46 48 46 49 47 49
47 51 48 49 48 50 49 50 49 51
50 51 50 52 50 54 51 52 51 53
52 53 52 54 52 56 53 55 53 56
53 57 54 56 54 60 55 57 55 58
56 57 56 60 56 62 57 58 57 61
57 62 57 63 58 59 58 61 59 61
60 62 60 64 61 63 61 65 62 63
62 64 62 65 62 66 63 65 64 66
65 66

```

(2) GPS, BANDRED1.FOR

Bandbreitenreduktion der Matrix A gegeben durch MATR61N.DAT \rightarrow Permutationsvektor PERM61N.DAT zur Umnummerierung

PERM61N.DAT

```

65      9

66
11      18      12      7      14      4      8      3      13      1
 2       6      10      5      17      9      15      16      22      23
30      24      31      33      32      38      40      41      42      39
46      48      47      52      51      57      58      56      63      64
65      66      59      61      62      60      55      53      54      49
50      45      44      43      37      36      35      29      21      34
27      28      26      25      19      20

```

Wir führen nun mit dem Permutationsvektor die Neunummerierung der Knoten aus und betrachten die Eigenschaften der permutierten Matrix $A' = P^T A P$ sowie die Stufenstruktur.

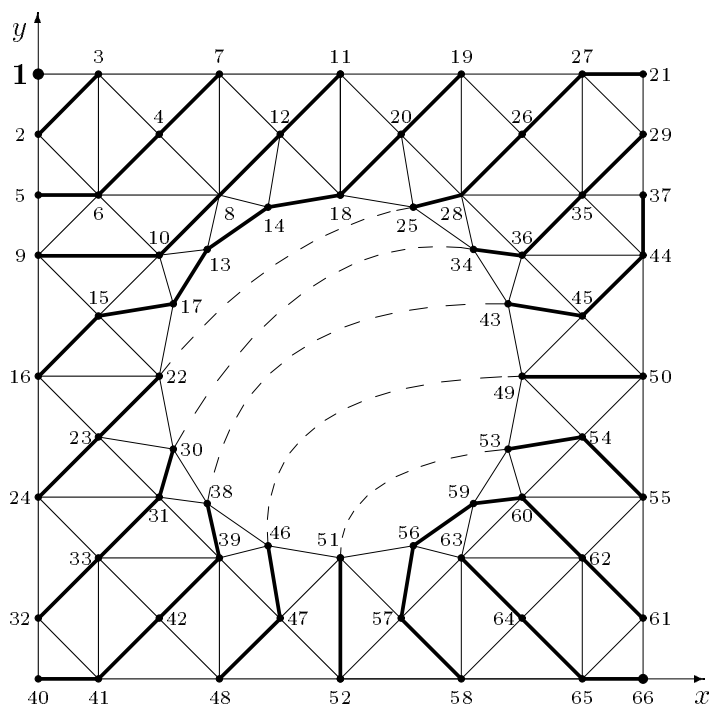


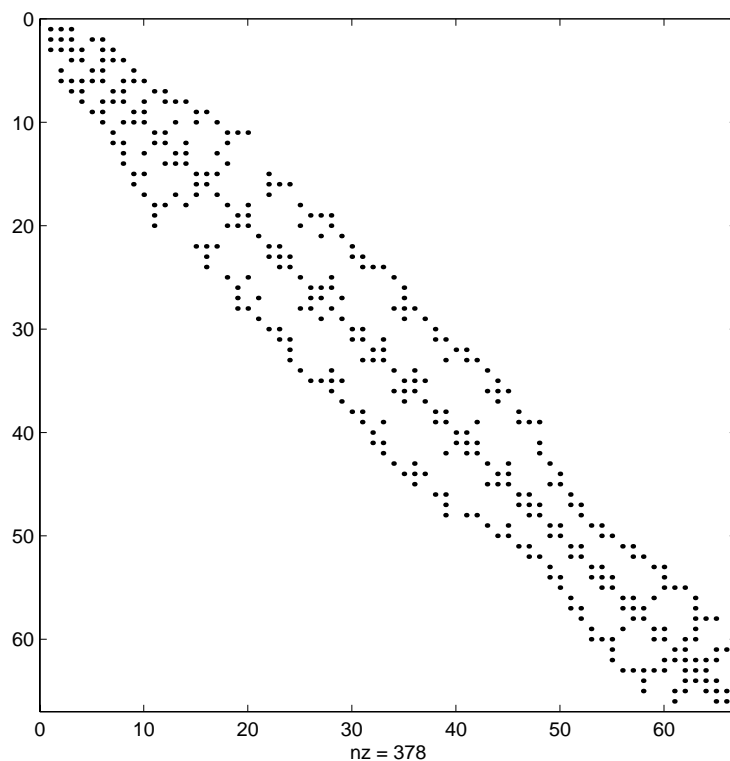
Abb. 1.14 Netz nach Neunummerierung der 66 Knoten, Startknoten = 10 \rightarrow 1

Die Stufenstruktur mit der Wurzel 1 und den 12 Stufen ist in der Abbildung fett hervorgehoben. Diese Stufen müssen sich wegen des Ringgebiets einmal “verzweigen“ und dann am rechten unteren Ende wieder zusammenfinden.

1. Stufe, Wurzel	1
2. Stufe (1 Zweig)	2,3
3. Stufe	5,6,4,7
4. Stufe	9,10,8,12,11
5. Stufe	16,15,17,13,14,18,20,19
6. Stufe (2 Zweige)	24,23,22 25,28,26,27,21
7. Stufe	32,33,31,30 34,36,35,29
8. Stufe	40,41,42,39,38 43,45,44,37
9. Stufe	48,47,46 49,50
10. Stufe	52,51 53,54,55
11. Stufe (1 Zweig)	58,57,56,59,60,62,61
12. Stufe	63,64,65,66

Tab. 1.6 Stufenstruktur mit Knoten in den Stufen beim GPS-Algorithmus

Wir stellen noch die Besetzungsstruktur der permutierten Matrix A' in MATLAB dar.



Die Bandbreiten von A' sind $bw = 1 + 2 \cdot 9 = 19$ und $b2 = 9$. Die zweite Bandbreite $b2$ ist natürlich kleiner als die maximale Knotennummerndifferenz in benachbarten Stufen, die hier $16 = 45 - 29$ beträgt (siehe 7. und 8. Stufe). Aber die Knoten 29 und 45 sind keine Nachbarknoten.

Mit den vielen Knotenpaaren (11,20), (19,28), (24,33), (25,34), (26,35), (32,41), (33,42), (34,43), (35,44), (36,45) und (39,48) aus benachbarten Stufen wird die Bandbreite $b2$ gebildet (siehe Abb. 1.14).

Es ist zu erwarten, dass $KNEW = 9$ die minimale Bandbreite ist.

(3) CM/RCM, CUTHILL.FOR

Rechnungen dazu liegen im Beispiel [65] Bsp. 1.5 vor.

Mit der Eingabedatei BEISP61N.DAT wurden mehrere Knoten als Startknoten getestet und folgende Ergebnisse erzielt.

1. Die minimale Bandbreite $b2(A') = 9$ wird für die Startknoten 22 und 37 erreicht.

MINIMALE BANDBREITE M = 9 FUER STARTPUNKT 22
DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

41	40	34	33	32	27	25	28	24	35	26	20	18	19
17	12	11	4	10	3	9	1	6	5	2	15	7	8
13	14	21	16	22	23	29	30	31	36	37	38	39	45
42	46	44	43	52	50	51	57	58	64	65	59	66	60
61	63	62	53	56	54	55	47	49	48				

MINIMALE BANDBREITE M = 9 FUER STARTPUNKT 37
DER PERMUTATIONSVEKTOR DER NEUNUMERIERUNG LAUTET :

58	57	64	65	59	62	60	63	53	66	61	56	54	55
46	48	47	40	39	34	32	33	26	25	28	24	27	20
19	18	17	13	12	5	11	4	1	10	6	2	3	8
14	9	7	15	16	21	22	29	23	31	30	35	36	37
38	42	50	41	45	43	44	49	52	51				

2. Der Startpunkt 10 bringt im CM nur die Bandbreite $b2 = 11$, während er bei GPS die kleinere Bandbreite 9 liefert.

Die Neunummerierung bei CM in Abbildung [65] Abb. 1.12 ist nicht so günstig. Insbesondere auf den mittleren Stufen gibt es dort sichtbare Abweichungen mit den entsprechenden Auswirkungen auf $b2$ (vgl. Tabellen [65] Tab. 1.2 sowie Tab. 1.6).

Literaturverzeichnis

- [1] Kielbasiński, A.; Schwetlick, H.: *Numerische lineare Algebra*. Mathematik für Naturwissenschaft und Technik Band 18, DVW, Berlin 1988.
- [2] Hackbusch, W.: *Iterative Lösung großer schwach besetzter Gleichungssysteme*. Leitfäden der angewandten Mathematik und Mechanik Band 69. B. G. Teubner Stuttgart 1991.
- [3] Maess, G.: *Vorlesungen über numerische Mathematik*. Band 1, 2. Akademie-Verlag Berlin 1984, 1988.
- [4] Schwarz, H. R.: 1. *Methode der finiten Elemente*. Leitfäden der angewandten Mathematik und Mechanik Band 47. B. G. Teubner Stuttgart 1991.
2. *FORTRAN-Programme zur Methode der finiten Elemente*. B. G. Teubner Stuttgart 1991.
- [5] Zlatev, Z.: *Computational Methods for General Sparse Matrices*. Math. and Its Appl. Vol.65. Kluwer Academic Publishers London 1991.
- [6] Gustavson, F.: *A Survey of Some Sparse Matrix Theory and Techniques*. Jahrbuch Überblicke Mathematik. B.I.-Wissenschaftsverlag Mannheim 1981.
- [7] Schmauder, M.; Weiss, R.; Schönauer, W.: *The CADSOL Program Package* (Version 1.1). Interner Bericht Nr. 46/92, RZ der Universität Karlsruhe 1992.
- [8] Schwetlick, H.; Kretschmar, H.: *Numerische Verfahren für Naturwissenschaftler und Ingenieure*. Fachbuchverlag Leipzig 1991.
- [9] Bramdler, A.; Allan, R. N.; Hamann, Y. M.: *Sparsity*. Pitman Publishing London 1976.
- [10] Schendel, U.: *Sparse Matrizen*. Oldenbourg Verlag München/Wien 1976.
- [11] Köckler, N.: *Numerische Algorithmen in Softwaresystemen : unter besonderer Berücksichtigung der NAG-Bibliothek*. B. G. Teubner Stuttgart 1990.
- [12] Rice, J. R.: *Numerical Methods, Software and Analysis*. 2nd Edition. Academic Press Inc. Boston 1993.
- [13] Govaerts, W.; Pryce, J. D.: *Mixed block elimination for linear systems with wide borders*. IMA Journ. of Numerical Analysis (1993)13, 161-180.
- [14] Collins, R. J.: *Bandwidth Reduction by Automatic Renumbering*. Int. Journ. Num. Methods in Engineering 6(1973) 345-356.
- [15] Gibbs, N. E.; Poole, W. G.; Stockmeyer, P. K.: *An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix*. SIAM Journ. Numerical Analysis 13(1976)2, 236-250.

- [16] Berger, C.: *Entwurf und Implementierung dünn besetzter Blockmatrizen in C++*. Diplomarbeit TU München IfI 1994.
- [17] Spiess, J.: *Untersuchungen des Zeitgewinns durch neue Algorithmen zur Matrix-Multiplikationen*. Computing 17, 23-36 (1976).
- [18] Neundorf, W.; Ortlepp, T.: *Berechnung von Matrix-Multiplikationen auf dem PC*. Preprint No. M 15/95, TUI Ilmenau IfMath August 1995.
- [19] Neundorf, W.: *Pascal-Programm INV_AUSP.PAS* .
Invertierung einer quadratischen Matrix mittels Austauschverfahren mit Spaltenpivotsuche und Zeilenvertauschung (Gauß-Jordan) sowie mit Pointertechnik. TU Ilmenau 1995.
- [20] Neundorf, W.; Hohlbein, D.: *Pascal-Programm MATVEK1.PAS* .
Kompaktspeicherung einer Matrix und Anwendung bei Matrix-Vektor-Multiplikation. TU Ilmenau 1996.
- [21] Samarskij, A. A.: *Theorie der Differenzenverfahren*. Akademische VG Geest & Portig K.-G. Leipzig 1984.
- [22] Neundorf, W.: *Behandlung großer Matrizen auf dem PC*. Preprint No M 11/96 Juni 1996 IfMath TU Ilmenau.
- [23] Neundorf, W.: *Manipulation von Matrizen I*. Preprint No M 16/96 November 1996 IfMath TU Ilmenau.
- [24] Neundorf, W.; Ortlepp, T.: *Unvollständige LU-Zerlegung und approximative Inverse von Blocktridiagonalmatrizen*. Preprint No M 5/97 Juni 1997 IfMath TU Ilmenau.
- [25] Neundorf, W.: *MATLAB - Teil I. Vektoren, Matrizen und lineare Gleichungssysteme*. Preprint No M 20/99 Juli 1999 IfMath TU Ilmenau.
- [26] Neundorf, W.: *MATLAB - Teil II. Speicheraspekte, spezielle LGS, SVD, EWP, Graphik, NLG, NLGS*. Preprint No M 23/99 September 1999 IfMath TU Ilmenau.
- [27] Neundorf, W.: *MATLAB - Teil III. Komplexe LGS, Interpolation, Splines*. Preprint No M 10/00 Mai 2000 IfMath TU Ilmenau.
- [28] Meister, A.: *Numerik linearer Gleichungssysteme*. Eine Einführung in moderne Verfahren. Friedr. Vieweg & Sohn VG mbH, Braunschweig 1999.
- [29] Meis, Th.; Marcowitz, U.: *Numerische Behandlung partieller Differentialgleichungen*. Springer-Verlag Berlin 1978.
- [30] Slavkovsky, P.; Rüde, U.: *Schnellere Berechnung klassischer Matrix-Multiplikationen*. Preprint TUM-I9032, SFB-Bereich Nr. 342/17/90 A, München September 1990.
- [31] Bonk, T.; Rüde, U.: *Performance Analysis and Optimization of Numerically Intensive Programs*. Preprint TUM-I9238, SFB-Bereich Nr. 342/26/92 A, München November 1992.
- [32] Stoer, J.: *Einführung in die Numerische Mathematik. Band 1*. Springer-Verlag Berlin 1979, 1989.

- [33] Spiess, J.: *Untersuchungen des Zeitgewinns durch neue Algorithmen zur Matrix-Multiplikationen*. Computing 17, 23-36 (1976).
- [34] Ortlepp, T.: *Schnellere Berechnung der klassischen Matrix-Multiplikation auf PC*. Beleg IfMath TU Ilmenau 1995.
- [35] Schwarz, H. R.: *Numerische Mathematik*. B. G. Teubner Stuttgart 1988.
- [36] Donner, K.: *Skalierung von Matrizen und numerische Stabilität der Gauß-Elimination*. Preprint Universität Passau, MIP-9514 September 1995.
- [37] Bauer, F. L.: *Optimally scaled matrices*. Numer. Mathematik 5(1963)73-87.
- [38] Wilkinson, J. H.; Reinsch, C.: *Linear Algebra*. Handbook for automatic computation, Vol. II. Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, Bd. 186. Berlin-Heidelberg-New York 1971.
- [39] Bruaset, A. M.: *A survey of preconditioned iterative methods*. Pitman Research Notes in Mathematics Series 328. Longman Scientific & Technical Essex, John Wiley & Sons, Inc., New York 1995.
- [40] Schaback, R; Werner, H.: *Numerische Mathematik*. Springer-Verlag Berlin 1993.
- [41] Überhuber, C.: *Computer-Numerik 1,2*. Springer-Verlag Berlin 1995.
- [42] Deuffhard, P.; Hohmann, A.: *Numerische Mathematik*. De Gruyter-Verlag Berlin New York 1991.
- [43] Ralston, A.: *A First Course in Numerical Analysis*. McGraw-Hill New York 1965.
- [44] Cherkasova, M. P.: *Collected Problems in Numerical Methods*. Akademie-Verlag Berlin 1972.
- [45] Schwarz, H. R.; Rutishauser, H.; Stiefel, E.: *Numerik symmetrischer Matrizen*. Leitfäden der angewandten Mathematik, Bd. 11. Stuttgart 1968, B. G. Teubner VG Leipzig 1969.
- [46] Jankowska, J.; Jankowski, M.: *Przegląd metod i algorytmów numerycznych*. Band 1. WNT Warszawa 1981.
- [47] Hämmerlin, G.; Hoffmann, K.-H.: *Numerische Mathematik*. Grundwissen Mathematik 7. Springer-Verlag Berlin 1991.
- [48] Stoer, J.; Burlisch, R.: *Einführung in die Numerische Mathematik II*. 3. Aufl. Springer-Verlag Berlin 1990.
- [49] Axelsson, O.: *Iterative Solution Methods*. Cambridge University Press 1994.
- [50] Engeln-Müllges, G.; Reutter, F.:
1. *Formelsammlung zur Numerischen Mathematik mit FORTRAN 77-Programmen*. Bibliogr. Institut Mannheim 1988.
2. *Formelsammlung zur Numerischen Mathematik mit Turbo Pascal-Programmen*. BI-Wissenschaftsverlag Mannheim 1991.
- [51] Engeln-Müllges, G.; Reutter, F.: *Numerik-Algorithmen mit ANSI C-Programmen*. (auch für Turbo Pascal, FORTRAN). BI-Wissenschaftsverlag Mannheim 1993.

- [52] Kose, K.; Schröder, R.; Wieliczek, K.: *Numerik sehen und verstehen*. Ein kombiniertes Lehr- und Arbeitsbuch mit Visualisierungssoftware. Vieweg Braunschweig 1992.
- [53] Zurmühl, R.; Falk, S.: *Matrizen und ihre Anwendungen*. Teil 2, Numerische Methoden. Springer-Verlag Berlin 1984.
- [54] Dietel, J.: *Formelsammlung zu Numerischen Mathematik mit Turbo Pascal-Programmen* (TPNUM). Rechenzentrum der RWTH Aachen 1993.
- [55] Plato, R.: *Numerische Mathematik kompakt*. Grundlagenwissen für Studium und Praxis. Vieweg Wiesbaden 2000.
- [56] Cuthill, E.: *Several strategies for reducing the band width of matrices*. In: Rose, D. J.; Willoughby, R.A. (ed.): *Sparse matrices and their applications*. Plenum, New York 1972, 157-166.
- [57] Cuthill, E.; McKee, J.: *Reducing the bandwidth of sparse symmetric matrices*. In: Proc. ACM Nat. Conf., New York 1969, 157-172.
- [58] Liu, W. H.; Sherman, A. H.: *Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee algorithms for sparse matrices*. SIAM J. Numer. Anal. 13 (1976) 198-213.
- [59] King, I. P.: *An automatic reordering scheme for simultaneous equations derived from network systems*. Intern. J. Numer. Meth. Engrg. 2 (1970) 523-533.
- [60] Hendrich, U.: *Über das Bandbreitenproblem für Produkte zweier Graphen*. Diplomarbeit IfMath TU Ilmenau 1989.
- [61] Rosen, R.: *Matrix bandwidth minimization*. Proc. 23rd ACM National Conf., 585-595, 1968.
- [62] Lierz, W.: *Lösung von großen Gleichungssystemen mit symmetrischer schwach besetzter Matrix*. Diplomarbeit Universität Köln 1975.
- [63] Neundorf, W.: *Wissenschaftliches Rechnen - Matrizen und lineare Gleichungssysteme*. Vorlesungsskript IfMath der TU Ilmenau, August 2002.
- [64] Neundorf, W.: *Bandbreitenreduktion - Teil 1*. Grundlagen, sparse Matrizen und ihre Verarbeitung. Preprint No M 06/02 September 2002 IfMath TU Ilmenau.
- [65] Neundorf, W.: *Bandbreitenreduktion - Teil 2*. Algorithmus von Cuthill-McKee. Preprint No M 07/02 September 2002 IfMath TU Ilmenau.

Anschrift:

Dr. rer. nat. habil. Werner Neundorf
Technische Universität Ilmenau, Institut für Mathematik
PF 10 05 65
D - 98684 Ilmenau
E-mail : werner.neundorf@tu-ilmenau.de
Homepage : http://www.mathematik.tu-ilmenau.de/~neundorf/index_de.html